



**UNIVERSIDAD
DE GRANADA**

DEPARTAMENTO DE MECÁNICA DE ESTRUCTURAS E
INGENIERÍA HIDRÁULICA

CÁLCULO MATRICIAL DE ESTRUCTURAS

Fernando Ávila; Esther Puertas;
Alejandro Martínez-Castro; Rafael Gallego

Primera edición
Marzo 2021

Cálculo Matricial de Estructuras

Fernando Ávila

Esther Puertas

Alejandro Martínez-Castro

Rafael Gallego

Departamento de Mecánica de Estructuras e Ingeniería Hidráulica

E.T.S. Ingeniería de Caminos, Canales y Puertos

Av. Fuentenueva s/n, 18071 Granada (España)

Universidad de Granada

Primera edición. Marzo 2021.

Obra protegida bajo licencia [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)



Núm. Reg. Propiedad Intelectual: 04/2021/4285

Índice general

Índice de figuras	IX
Introducción	XI
1. Conceptos básicos del Análisis Matricial	1
1.1. Introducción al Cálculo Matricial	1
1.2. Discretización y grados de libertad	2
1.3. Métodos de Equilibrio y Compatibilidad	3
1.4. Concepto de matriz de rigidez y matriz de flexibilidad	5
2. Sistemas de coordenadas y matrices de rigidez	17
2.1. Introducción	17
2.2. Sistemas de coordenadas	18
2.3. Matrices de rigidez elementales	24
2.4. Transformación de coordenadas	28
3. El Método Directo de la Rigidez	43
3.1. Planteamiento del problema	43
3.2. El Método Directo de la Rigidez	46
3.3. Aplicación de las condiciones de contorno y resolución del sistema	49
3.4. Cálculo de esfuerzos y reacciones	50
3.5. Propiedades de la matriz de rigidez	53
3.6. Pasos para la resolución de problemas	54
4. Cargas aplicadas en barras y otras acciones	67
4.1. Introducción	67
4.2. Cargas aplicadas en barras	68
4.3. Cargas térmicas	72
4.4. Asientos en apoyos	73

5. Técnicas complementarias y casos particulares	89
5.1. Introducción	89
5.2. Condensación estática y libertades en elementos	90
5.3. Apoyos no concordantes	111
5.4. Apoyos elásticos	114
5.5. Ligaduras entre grados de libertad	117
5.6. Subestructuras	124
5.7. Elementos especiales	125
Anexo A. Matrices de rigidez elementales	131
A.1. Estructura articulada plana (A2D)	131
A.2. Pórtico plano (P2D)	132
A.3. Emparrillado plano (E2D)	133
Anexo B. Matrices de rigidez elementales con gdl condensados	135
B.1. Pórtico plano con libertad de giro en el extremo inicial	135
B.2. Pórtico plano con libertad de giro en el extremo final	137
B.3. Pórtico plano con libertad de desplazamiento en el extremo inicial	138
B.4. Pórtico plano con libertad de desplazamiento en el extremo final	139
Anexo C. Glosario	141

Índice de figuras

1.1. Discretización.	2
1.2. Grados de libertad en pórtico bidimensional.	3
1.3. Diagrama de Tonti.	4
1.4. Comportamiento de un muelle de rigidez K sometido a una fuerza F	5
1.5. Ejemplo de barra con 6 gdl.	7
1.6. Ejercicio 1.1. Barra en voladizo.	8
1.7. Ejercicio 1.1. Desplazamiento unitario en gdl 1.	9
1.8. Ejercicio 1.1. Desplazamiento unitario en gdl 2.	9
1.9. Ejercicio 1.1. Desplazamiento unitario en gdl 3.	9
1.10. Ejercicio 1.2. Estructura de barras.	10
1.11. Ejercicio 1.2. Desplazamiento horizontal unitario.	10
1.12. Ejercicio 1.2. Desplazamiento vertical unitario.	10
1.13. Ejercicio 1.2. Giro unitario	11
1.14. Ejercicio Python 1.1. Reacción vertical en el empotramiento en función de la longitud de la viga.	15
2.1. Sistema de coordenadas de referencia.	18
2.2. Estructura articulada plana.	18
2.3. Estructura articulada tridimensional.	19
2.4. Pórtico plano.	19
2.5. Pórtico tridimensional.	20
2.6. Emparrillado	20
2.7. Elemento en estructura articulada plana.	21
2.8. Elemento en estructura articulada tridimensional.	21
2.9. Elemento en pórtico plano.	22
2.10. Elemento en pórtico tridimensional.	22
2.11. Elemento en emparrillado.	23
2.12. Sistema de coordenadas nodal en apoyo no concordante.	24
2.13. Elemento en A2D o A3D, desplazamiento unitario en el gdl 1.	25
2.14. Elemento en A2D o A3D, desplazamiento unitario en el gdl 2.	25
2.15. Elemento en P2D, desplazamientos unitarios en los 3 primeros gdl.	25

2.16. Transformación de coordenadas nodal-global.	28
2.17. Transformación de coordenadas local-global para A2D.	30
2.18. Transformación de coordenadas local-global para P2D.	32
2.19. Ejercicio 2.1. Elemento P2D.	33
2.20. Ejercicio 2.1. Fuerzas en la barra.	35
2.21. Ejercicio 2.1. Esfuerzos en la barra.	35
3.1. Grados de libertad de P2D en los sistemas global y local.	45
3.2. Diagrama de Tonti para la estructura discretizada.	45
3.3. Tipos de apoyos comunes en estructuras planas.	49
3.4. Ejercicio 3.1. Pórtico 2D.	51
3.5. Ejercicio 3.1. Discretización y sistemas de coordenadas.	51
3.6. Ejercicio 3.2. Estructura planteada.	56
3.7. Ejercicio 3.2. Discretización y sistema de coordenadas.	57
3.8. Ejercicio 3.2. Fuerzas en los nudos de la barra a	60
3.9. Ejercicio 3.2. Diagramas de esfuerzos en la barra a	60
3.10. Ejercicio 3.2. Fuerzas en los nudos de la barra b	60
3.11. Ejercicio 3.2. Diagramas de esfuerzos en la barra b	61
3.12. Ejercicio 3.2. Fuerzas nodales y diagrama de esfuerzos en la barra c	61
3.13. Ejercicio 3.2. Reacciones y cargas en la estructura.	62
3.14. Ejercicio Python 3.1. Estructura P2D.	62
3.15. Ejercicio Python 3.1. Estructura P2D - Variante propuesta.	66
4.1. Descomposición de la estructura en estados I y II.	68
4.2. Viga biempotrada con carga uniforme inclinada.	69
4.3. Descomposición de carga uniforme inclinada.	70
4.4. Fuerzas de empotramiento en coordenadas globales.	70
4.5. Fuerzas de empotramiento resultantes	71
4.6. Equivalencia con barra de longitud $L \cos \alpha$	71
4.7. Barra sometida a gradiente térmico lineal en su sección.	72
4.8. Fuerzas de empotramiento en barra sometida a gradiente tér- mico lineal en su sección.	73
4.9. Fuerzas de empotramiento en barra sometida a carga térmica constante en su sección.	73
4.10. Estructura con asiento en apoyo.	74
4.11. Ejercicio 4.1. Pórtico con acciones en barras.	75
4.12. Ejercicio 4.1. Sistemas de coordenadas y $gd1$	75
4.13. Ejercicio 4.1. Fuerzas de empotramiento en barra a	77
4.14. Ejercicio 4.1. Fuerzas de empotramiento en barra b	77
4.15. Ejercicio 4.1. Fuerzas en los nudos de la barra b	79
4.16. Ejercicio 4.1. Diagramas de esfuerzos en la barra b	80
4.17. Ejercicio Python 4.1. Estructura con carga distribuida.	80
5.1. Ejercicio 5.1. Barra P2D con carga uniforme.	92

5.2. Ejercicio 5.1. gdl de la barra antes y después de condensar θ_2 .	92
5.3. Ejercicio 5.1. Reacciones en barra empotrada-apoyada con carga uniforme.	94
5.4. Condensación en barras unidas por rótula.	96
5.5. Condensación en barras unidas por rótula con momento en nudo.	96
5.6. Condensación en barras unidas con deslizadera	96
5.7. Ejemplo de condensación en pórtico	97
5.8. Ejemplo de condensación en pórtico (II)	97
5.9. Ejercicio 5.2. Estructura.	98
5.10. Ejercicio 5.2. gdl de la estructura condensados u_{1x} y θ_3	99
5.11. Ejercicio 5.2. Fuerzas la barra b	101
5.12. Ejercicio 5.2. Diagramas de esfuerzos en la barra b	101
5.13. Ejercicio Python 5.1. Estructura.	102
5.14. Estructura con apoyo no concordante.	112
5.15. Apoyo elástico en sus 3 gdl	115
5.16. Ejercicio 5.3. Estructura con apoyo elástico.	116
5.17. Apoyo elástico no concordante con el sistema de coordenadas global.	117
5.18. Ejemplo de ligadura entre gdl : pórtico con cimentación en losa.	118
5.19. Ejercicio 5.1. Estructura con sólido rígido.	119
5.20. Ejercicio 5.1. Separación SR – estructura de barras.	120
5.21. Ejercicio 5.1. gdl en la estructura de barras (cond. θ_1 , θ_4 , y u_{5x}).	120
5.22. Estructura compuesta por subestructuras.	124
5.23. Barra con rótula en su punto medio, con 3 gdl en cada nudo.	125
5.24. Elementos P2D unidos por una rótula, con los giros condensados en esa unión.	126
5.25. Barra con rótula en cualquier punto intermedio.	128
5.26. Elemento con extremos flexibles.	129
5.27. Elemento con extremos rígidos.	130

Introducción

Con la presente obra, los autores buscan introducir al lector en la metodología del Cálculo Matricial para la resolución de problemas estructurales. A lo largo de las siguientes páginas, se describen en detalle los conceptos básicos del Cálculo Matricial y su aplicación para la resolución de estructuras, así como diferentes técnicas que permitirán resolver estructuras de mayor complejidad introduciendo diversas condiciones particulares de carga y apoyo. Para la correcta comprensión de los temas descritos en esta obra es conveniente tener unos mínimos conocimientos previos sobre análisis matricial y de resistencia de materiales.

El texto, que pretende ser una herramienta útil para la aplicación del Cálculo Matricial de estructuras, combina explicaciones teóricas y demostraciones numéricas con diversos ejercicios prácticos que muestran la aplicación de los conceptos. Se incluyen también ejercicios resueltos con el lenguaje de programación Python¹, con cada vez un mayor peso en los ámbitos de la Ciencia y la Ingeniería, describiendo el procedimiento seguido para que pueda ser comprensible incluso por lectores que no estén iniciados en el uso de este lenguaje.

De este modo, esta obra se divide en los siguientes capítulos:

1. **Conceptos básicos del Análisis Matricial.** Se introducen los conceptos esenciales que se deben conocer para aplicar el método matricial de cálculo de estructuras, tales como discretización, grados de libertad, equilibrio y compatibilidad, y rigidez y flexibilidad.
2. **Sistemas de coordenadas y matrices de rigidez.** Se expone la importancia de definir y trabajar en el sistema de coordenadas adecuado, y se definen las matrices de rigidez de los distintos tipos de elementos más comunes, así como el modo en que cambiar de un sistema de coordenadas a otro.
3. **El Método Directo de la Rigidez.** Este será el método que se aplicará en este texto para resolver los problemas estructurales. Se describen las bases del mismo y el proceso a seguir para la resolución de problemas.

¹Disponibles en el repositorio https://github.com/rafaelmgallego/CALMAT_Python

4. **Cargas aplicadas en barras y otras acciones.** En este capítulo se describe el procedimiento a realizar cuando la estructura que se desea resolver tiene acciones aplicadas en las barras o sufre asientos en los apoyos.
5. **Técnicas complementarias y casos particulares.** Se ofrece al lector herramientas para resolver estructuras que presenten condiciones particulares, como apoyos no concordantes, apoyos elásticos o ligaduras entre grados de libertad. Además, se expone el método de condensación estática de grados de libertad, que permite reducir el tamaño del problema, condensando las relaciones matriciales a un subconjunto de los grados de libertad.

Finalmente, en la sección de Anexos, se incluyen las matrices de rigidez elementales más empleadas, incluidas aquellas con grados de libertad condensados. Se incluye también un breve glosario de la terminología empleada en este texto.

Capítulo 1

Conceptos básicos del Análisis Matricial

Índice

1.1. Introducción al Cálculo Matricial	1
1.2. Discretización y grados de libertad	2
1.3. Métodos de Equilibrio y Compatibilidad	3
1.4. Concepto de matriz de rigidez y matriz de flexibilidad	5

1.1. Introducción al Cálculo Matricial

En el ámbito de la ingeniería, podemos definir **estructura** como un conjunto de elementos resistentes capaces de mantener su forma y cualidades a lo largo del tiempo, bajo la acción de las cargas y agentes exteriores a que han de estar sometidos¹. Es posible clasificar las estructuras según diferentes criterios; si atendemos a la tipología de los elementos que las componen y de las uniones entre los mismos, podemos distinguir:

- Estructuras de elementos barra.
 - Articuladas.
 - De nudos rígidos bidimensionales (pórticos y emparrillados).
 - De nudos rígidos tridimensionales.
- Estructuras de elementos superficiales
 - Superficie media plana: placas y láminas.

¹[Diccionario de la Real Academia de Ingeniería.](#)

- Superficie media curva: láminas.
- Sólidos deformables tridimensionales.

La expresión *Cálculo Matricial* se refiere a las metodologías generales para el cálculo de **estructuras de barras**, basadas en la nomenclatura algebraica de matrices y vectores, y orientadas a la implementación en ordenador. Las estructuras superficiales planas o curvas, y los sólidos tridimensionales se podrán resolver, de manera aproximada, mediante el Método de los Elementos Finitos (MEF), que es una extensión del *Método Directo de la Rigidez* que se presenta en este libro.

Los métodos matriciales para el cálculo de estructuras son una extensión de las ideas desarrolladas por Maxwell y Mohr en el siglo XIX. Estos métodos presentan, en general, una gran simplicidad, bastando para su aplicación el conocimiento de los teoremas generales del cálculo de estructuras y una base de álgebra vectorial y matricial. El uso de ordenadores cada vez más potentes ha permitido implementar métodos de cálculo matriciales de carácter general para resolver una gran variedad de problemas, pudiéndose ampliar con facilidad el código para introducir nuevos elementos o condiciones. Es importante, sin embargo, conocer los fundamentos básicos de la metodología para no perder el sentido físico del problema.

1.2. Discretización y grados de libertad

Para el análisis de la estructura partimos de los conocimientos que aporta la **Resistencia de Materiales**, que nos permite obtener los resultados para una barra sometida a condiciones cualesquiera de carga y apoyo. Estas barras o, de forma más general, **elementos** se unen entre sí en una serie de puntos llamados **nudos**.

La discretización consiste en representar la estructura como un ensamblaje de *barras*, que se unen en los *nudos*. El comportamiento global de la estructura lo encontraremos estableciendo las ecuaciones de la mecánica estructural correspondientes en cada barra y en cada nudo, como veremos más adelante.

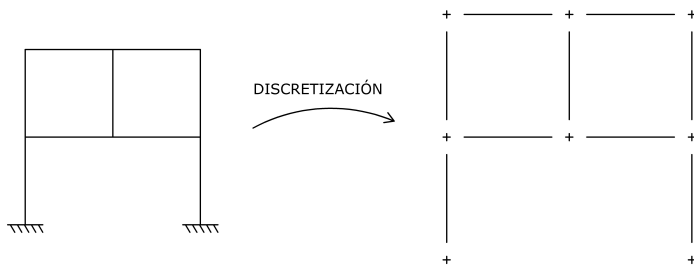
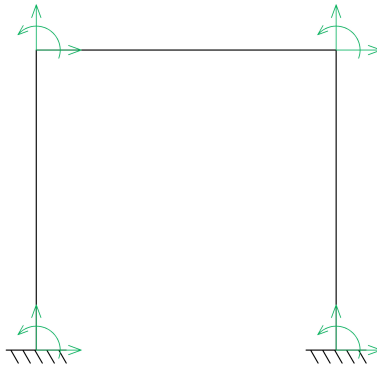


Figura 1.1: Discretización.

Se define como **grado de libertad** (en adelante **gdl**) de un punto al número de coordenadas que es preciso fijar para que su movimiento quede determinado. La idea de discretización descrita anteriormente implica que se pasa de tener un número infinito de **gdl** en la estructura (infinitos puntos) a un número finito (número finito de nudos).

El número de **gdl** se puede expresar para un nudo, para un elemento o para toda la estructura o sistema. En cada caso, existirán tantos desplazamientos/giros y fuerzas externas posibles como **gdl** haya.



$$3 \text{ gdl/nudo} \rightarrow 6 \text{ gdl/elemento} \rightarrow 12 \text{ gdl/estructura}$$

Figura 1.2: Grados de libertad en pórtico bidimensional.

1.3. Métodos de Equilibrio y Compatibilidad

En cualquier problema estructural se cumplen tres tipos de relaciones, que son a su vez las únicas condiciones de las que se dispone para la resolución de las estructuras:

- Las **ecuaciones de equilibrio** de esfuerzos en la estructura y cargas externas.
- Las **ecuaciones de compatibilidad** entre las deformaciones de los elementos y los desplazamientos de los nudos, y las condiciones de contorno.
- Las **leyes de comportamiento** entre esfuerzos y deformaciones.

Estas relaciones vienen esquematizadas en el llamado Diagrama de Tonti:

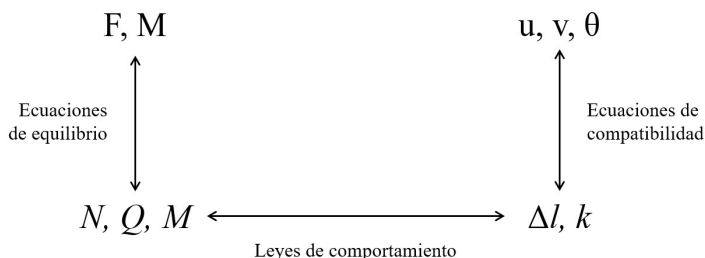


Figura 1.3: Diagrama de Tonti.

En el caso de las estructuras **isostáticas** el diagrama puede recorrerse paso a paso en sentido antihorario; pues, por definición de *isostatismo*, pueden calcularse los esfuerzos resolviendo las ecuaciones de equilibrio, obtener después las deformaciones en los elementos empleando las leyes de comportamiento y finalmente aplicar las ecuaciones de compatibilidad para obtener los desplazamientos a partir de dichas deformaciones.

Para **estructuras hiperestáticas**, sin embargo, no es posible aplicar de forma aislada las ecuaciones de equilibrio para obtener los esfuerzos, por lo que habrá que emplear de forma conjunta los tres conjuntos de ecuaciones. Esto se puede realizar de dos formas distintas, mediante lo que se denominan *Métodos de Compatibilidad* y *Métodos de Equilibrio*.

Recordémoslos sucintamente:

- **Métodos de Compatibilidad, de Fuerzas o de Flexibilidad:** se liberan una serie de ligaduras (restricciones en los $gd1$) hasta convertir la estructura en isostática, expresando las fuerzas y esfuerzos correspondientes a esas ligaduras como acciones externas desconocidas. A continuación se resuelve la estructura como isostática, aplicando sucesivamente las ecuaciones de equilibrio, comportamiento y compatibilidad. Las ecuaciones finales, las de compatibilidad que dan nombre al método, conforman un sistema de ecuaciones donde las fuerzas/esfuerzos hiperestáticos serán las incógnitas.
- **Método del Equilibrio, de los Desplazamientos o de la Rigidez:** tomando como incógnitas todos los *desplazamientos* de los nudos, se calculan las deformaciones en los elementos mediante las ecuaciones de compatibilidad, a partir de éstas los esfuerzos en las barras, y finalmente se plantean las ecuaciones de *equilibrio* que relacionan fuerzas externas y esfuerzos, obteniéndose un sistema de ecuaciones donde las incógnitas son los desplazamientos.

El *Método de la Flexibilidad* ha sido tradicionalmente usado en *Resistencia de Materiales* para resolver estructuras hiperestáticas de tamaño pequeño, teniendo en cuenta las características particulares de cada estructura para simplificar su resolución. El *Método de la Rigidez*, por el contrario, tiene la ventaja de permitir la definición del problema de forma general, sin

hacer elecciones dependientes de la estructura en estudio, aunque a veces esto implique la resolución de un mayor número de ecuaciones en el sistema. El *Método de Cross*, un método popular hasta finales del s.XX para la resolución de estructuras complejas como pórticos de edificios de gran altura, es un *Método de Equilibrio*, aunque al estar concebido para su aplicación manual, sin ordenadores, las ecuaciones de equilibrio se resuelven de manera iterativa de manera muy sistemática y con un mínimo de operaciones.

En este libro vamos a presentar el método más útil y fácil de programar, denominado **Método Directo de la Rigidez**. Como su nombre indica, es un método de equilibrio, por lo que las incógnitas del sistema de ecuaciones resultante serán los desplazamientos. Este método es que el implementan la totalidad de los programas comerciales de cálculo de estructuras, junto con la generalización del mismo, el *Método de los Elementos Finitos*.

1.4. Concepto de matriz de rigidez y matriz de flexibilidad

De forma intuitiva, los conceptos de rigidez y flexibilidad de una estructura o elemento, se refieren a la mayor o menor magnitud de los desplazamientos que se producen al aplicar unas cargas determinadas: los desplazamientos serán menores si la estructura es más rígida y mayores si esta es más flexible. Si tomamos como ejemplo un muelle sometido a una fuerza de tracción F que produce un alargamiento u , tenemos:

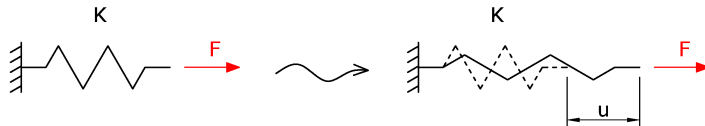


Figura 1.4: Comportamiento de un muelle de rigidez K sometido a una fuerza F .

Podemos expresar este comportamiento mediante la siguiente ecuación, conocida como ley de Hooke²:

$$F = k u$$

donde k es la constante de rigidez del muelle. La rigidez es igual a la fuerza necesaria para producir un desplazamiento unitario. También lo podemos expresar, de forma recíproca, como:

$$u = a F$$

²Publicada en 1673 como anagrama *ceiüinossttuv*, para asegurarse la prioridad del descubrimiento, que dos años describió en la frase latina, *Ut tension, sic vis*, que traducido al español sería *Como la extensión, así la fuerza*

empleando en este caso la constante de flexibilidad a , que evidentemente se relaciona con la de rigidez mediante $a = 1/k$. La flexibilidad a es igual al desplazamiento producido al aplicar una fuerza de valor unitario.

Estas ideas pueden extenderse a una estructura generalizando los conceptos de fuerza y desplazamiento. De este modo, si una estructura tiene n gdl, tendremos un conjunto de fuerzas F_1, F_2, \dots, F_n aplicadas sobre los gdl, y los correspondientes movimientos u_1, u_2, \dots, u_n . La ley de Hooke generalizada será el siguiente conjunto de ecuaciones lineales:

$$\begin{aligned} F_1 &= K_{11}u_1 + K_{12}u_2 + \dots K_{1n}u_n \\ F_2 &= K_{21}u_1 + K_{22}u_2 + \dots K_{2n}u_n \\ &\vdots \\ F_n &= K_{n1}u_1 + K_{n2}u_2 + \dots K_{nn}u_n \end{aligned}$$

o las relaciones recíprocas,

$$\begin{aligned} u_1 &= A_{11}F_1 + A_{12}F_2 + \dots A_{1n}F_n \\ u_2 &= A_{21}F_1 + A_{22}F_2 + \dots A_{2n}F_n \\ &\vdots \\ u_n &= A_{n1}F_1 + A_{n2}F_2 + \dots A_{nn}F_n \end{aligned}$$

siendo K_{ij} y A_{ij} constantes que dependen de las características elásticas de la estructura, como veremos más adelante.

Podemos escribir los sistemas de ecuaciones anteriores *matricialmente* definiendo un vector de fuerzas \mathbf{F} de tamaño $n \times 1$,

$$\mathbf{F} = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{pmatrix}$$

y un vector de desplazamientos,

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}$$

del mismo tamaño, de modo que las ecuaciones lineales anteriores pueden escribirse de forma compacta como,

$$\mathbf{F} = \mathbf{K}\mathbf{u}$$

y

$$\mathbf{u} = \mathbf{A}\mathbf{F}$$

respectivamente, siendo \mathbf{K} la *matriz de rigidez* y \mathbf{A} la *matriz de flexibilidad* de la estructura, ambas de dimensión $n \times n$.

Las matrices serán,

$$\mathbf{K} = \begin{pmatrix} K_{11} & K_{12} & \cdots & K_{1n} \\ K_{21} & K_{22} & \cdots & K_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ K_{n1} & K_{n2} & \cdots & K_{nn} \end{pmatrix}$$

y

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{pmatrix}$$

Las matrices de rigidez y flexibilidad cumplirán la relación,

$$\mathbf{A} = \mathbf{K}^{-1}$$

Veamos estas relaciones tomando como ejemplo sencillo el de una barra de pórtico bidimensional, con 3 gdl por nudo. En la figura los gdl están numerados según el orden en que vamos a colocarlos en los vectores de fuerzas y desplazamientos.



Figura 1.5: Ejemplo de barra con 6 gdl.

La ecuación matricial que relaciona fuerzas y desplazamientos en la barra mediante la matriz de rigidez será:³

³La línea discontinua en los vectores y matrices se usará para separar los términos que corresponden a nudos diferentes.

$$\mathbf{F} = \mathbf{K}\mathbf{u} \Rightarrow$$

$$\Rightarrow \begin{pmatrix} F_{1x} \\ F_{1y} \\ M_1 \\ F_{2x} \\ F_{2y} \\ M_2 \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & K_{56} \\ K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & K_{66} \end{pmatrix} \begin{pmatrix} u_{1x} \\ u_{1y} \\ \theta_1 \\ u_{2x} \\ u_{2y} \\ \theta_2 \end{pmatrix}$$

La definición de la **matriz de rigidez** es análoga a la descrita para la constante del muelle, es decir, es la matriz de fuerzas a aplicar para producir un desplazamiento unidad en el **gdl** correspondiente. Del mismo modo, la matriz de flexibilidad será la matriz de desplazamientos producidos al aplicar una fuerza unidad en el **gdl** correspondiente.

Con mayor precisión: la columna j de la matriz de rigidez \mathbf{K} representa las fuerzas que es necesario aplicar en cada **gdl** para producir una deformada en la que $u_j = 1$ y $u_k = 0$ para $k \neq j$.

De modo análogo, la columna j de la matriz de flexibilidad \mathbf{A} representa los desplazamientos que se producen en cada **gdl** cuando se aplica una fuerza $F_j = 1$ y $F_k = 0$ para $k \neq j$.

Veamos dos ejemplos prácticos para afianzar la comprensión del concepto de matriz de rigidez.

Ejercicio 1.1 Calcular la matriz de rigidez de una barra en voladizo con módulo de elasticidad E , área de la sección A e inercia de la sección I .

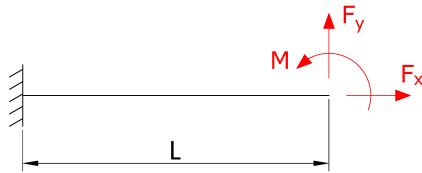


Figura 1.6: Ejercicio 1.1. Barra en voladizo.

Solución

Cada columna de la matriz \mathbf{K} corresponde con las fuerzas a aplicar en cada **gdl** para provocar un desplazamiento unidad en cada uno de los 3 **gdl** del extremo libre la barra. Por tanto, la primera columna corresponderá con $u_x = 1$, $u_y = \theta = 0$:

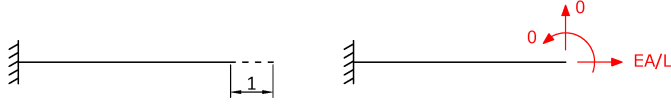


Figura 1.7: Ejercicio 1.1. Desplazamiento unitario en gdl 1.

Segunda columna, $u_x = 0$, $u_y = 1$, $\theta = 0$:

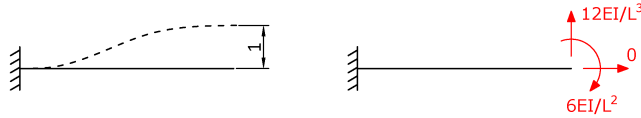


Figura 1.8: Ejercicio 1.1. Desplazamiento unitario en gdl 2.

Tercera columna, $u_x = 0$, $u_y = 0$, $\theta = 1$:

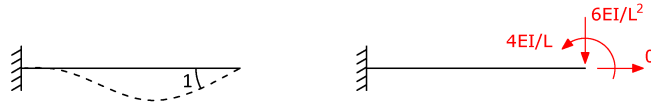


Figura 1.9: Ejercicio 1.1. Desplazamiento unitario en gdl 3.

Por tanto, se obtiene:

$$\mathbf{F} = \mathbf{K}\mathbf{u} \Rightarrow \begin{pmatrix} F_x \\ F_y \\ M \end{pmatrix} = \begin{pmatrix} \frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \\ \theta \end{pmatrix}$$

Como se ha indicado antes, la matriz de flexibilidad se podría obtener directamente como la inversa de la matriz de rigidez calculada ($\mathbf{A} = \mathbf{K}^{-1}$).

Ejercicio 1.2 En la siguiente estructura de barras, calcular los desplazamientos producidos por las cargas aplicadas.

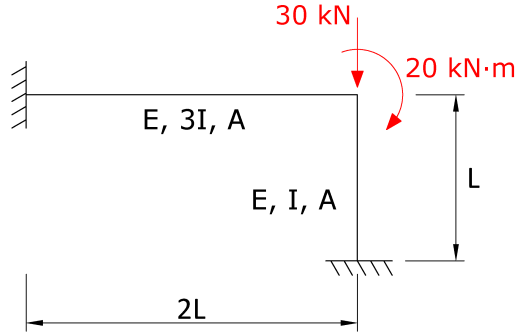


Figura 1.10: Ejercicio 1.2. Estructura de barras.

Datos: $E = 2 \cdot 10^8 \text{ kN/m}^2$, $I = 5 \cdot 10^{-5} \text{ m}^4$, $A = 5 \cdot 10^{-3} \text{ m}^2$, $L = 1 \text{ m}$.

Solución

El problema se resolverá mediante el método de la rigidez, calculando en primer lugar la matriz \mathbf{K} y resolviendo el sistema $\mathbf{F} = \mathbf{K}\mathbf{u}$.

Calculamos la matriz de rigidez aplicando desplazamientos unitarios en el nudo en cuestión. Primera columna de la matriz, desplazamiento horizontal unitario:

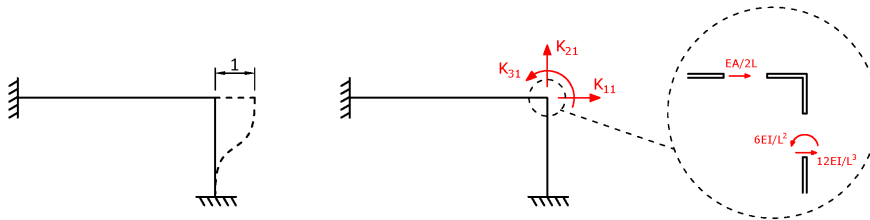


Figura 1.11: Ejercicio 1.2. Desplazamiento horizontal unitario.

Segunda columna, desplazamiento vertical unitario:

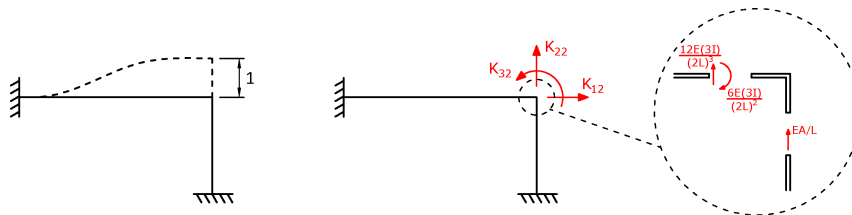


Figura 1.12: Ejercicio 1.2. Desplazamiento vertical unitario.

Tercera columna, giro unitario:

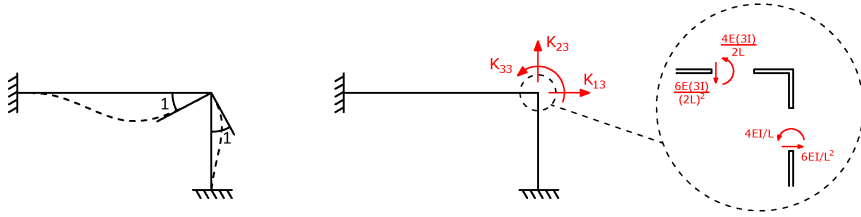


Figura 1.13: Ejercicio 1.2. Giro unitario

Formando la matriz \mathbf{K} y montando el sistema:

$$\mathbf{F} = \mathbf{K}\mathbf{u} \Rightarrow \begin{pmatrix} 0 \\ -30 \\ 20 \end{pmatrix} = \begin{pmatrix} \frac{2EA}{L} + \frac{12EI}{L^3} & 0 & \frac{6EI}{L^2} \\ 0 & \frac{EA}{L} + \frac{9EI}{2L^3} & -\frac{9EI}{2L^3} \\ \frac{6EI}{L^2} & -\frac{9EI}{2L^2} & \frac{10EI}{L} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \\ \theta \end{pmatrix}$$

Sustituyendo los valores de la variables y resolviendo el sistema:

$$u_x = 1.962 \cdot 10^{-5} \text{ m}$$

$$u_y = 1.998 \cdot 10^{-5} \text{ m}$$

$$\theta = 2.028 \cdot 10^{-4} \text{ rad}$$

Ejercicio Python 1.1

A continuación se presenta el primero de una serie de ejercicios resueltos mediante el lenguaje de programación **Python** que se incluyen en este texto. Existen varias distribuciones de **Python** disponibles de forma gratuita, los ejercicios incluidos en este libro están resueltos en la versión **Python 3**.

Le recomendamos instalar el entorno **Anaconda** (Continuum Analytics). [Descargue](#) la versión compatible con su sistema (**Windows**, **Linux**, **OS-X**) para 32 o 64 bits. Si dispone de un equipo con recursos limitados, puede instalar una versión reducida, denominada **Miniconda**. Dentro de este entorno, utilice (o instale si ha optado por **Miniconda**) el entorno **Spyder**, que tiene activados recursos de **IPython**, **numpy** y **matplotlib** para facilitar la tarea en cálculo científico.

En el presente ejercicio se presenta un código que permite calcular, mediante un producto matricial, las fuerzas existentes en los extremos de una viga a partir de los desplazamientos conocidos en los extremos.

A modo de ejemplo, se resuelve el caso de una viga metálica correspondiente a un perfil IPN-200 en voladizo. Tenga en cuenta que cualquier dato puede modificarse en el código para variar las características del perfil o sus condiciones de apoyo. Se toman, por tanto, los siguientes datos de entrada:

- Módulo de elasticidad: $E = 210 \text{ GPa}$
- Área: $A = 33.5 \text{ cm}^2$
- Inercia: $I = 2140 \text{ cm}^4$
- Longitud de la viga: $L = 2.8 \text{ m}$

Además, se consideran conocidos los desplazamientos en el extremo libre del voladizo:

- Desplazamiento horizontal: $u_{jx} = 1 \text{ mm}$
- Desplazamiento vertical: $u_{jy} = 1 \text{ mm}$
- Giro: $\theta_j = 1 \text{ mrad}$

El siguiente código calcula (y muestra) la matriz de rigidez de la viga, a través de una función definida a tal efecto; y luego opera para obtener el vector de fuerzas externas, pudiendo así conocer las fuerzas aplicadas en los extremos de la viga que dan lugar a los citados desplazamientos.

De forma adicional, se realiza un estudio paramétrico para observar la dependencia entre la longitud de la viga y la reacción en el empotramiento, manteniendo todos los demás parámetros constantes.

```
1 #=====
2 #Bibliotecas y funciones necesarias
3 #=====
4
5 from __future__ import division #Para no tener problemas en la
    división entre enteros
6 import numpy as np             #Para importar la librería numpy
7 import matplotlib.pyplot as plt #Para dibujar
8
9 #=====
10 #Función para crear la matriz de rigidez de un elemento de 6 gdl
11 #Pórtico 2D (P2D)
12 #=====
13 def k_p2d(ea, ei, lon):
14     k_local = np.zeros([6,6])
15
16     #Se rellenan primero los términos de la triangular superior
17     #En Python las matrices y vectores se indexan desde el
18     #0,1,2,3...
19     #Una matriz 6x6 tendrá índices desde el 0,1,2,3,4,5.
20     k_local[0,:] = np.array([ ea / lon, 0., 0., -ea / lon, 0.,
21                               0.])
22     k_local[1,1:] = np.array([12 * ei / lon**3, 6 * ei / lon**2,
23                               -12 * ei / lon**3, 6 * ei / lon**2])
```



```

22     k_local[2,2:] = np.array([4 * ei / lon, 0, -6 * ei / lon**2,
23                               2 * ei / lon ])
24     k_local[3,3:] = np.array([ea / lon, 0, 0])
25     k_local[4,4:] = np.array([12 * ei / lon**3, -6 * ei / lon
26                               **2])
27     k_local[5,5]  = 4 * ei / lon
28
29     #El resto de la matriz se construye por simetría.
30     #Se suma k_local + traspuesta (k_local)
31     #Se resta la diagonal de k_local para que no aparezca 2
32     veces
33     #Para escribir la traspuesta de k_local, simplemente se
34     escribe k_local.T
35     k_local = (k_local + k_local.T) - np.diag ( np.diag(k_local)
36     )
37
38     #Salida de la función:
39     return k_local
40
41 #=====
42 #Datos de entrada
43 #=====
44 #Sistema de unidades: [kN], [m]
45 #Datos de un IPN-200
46
47 Elast = 210E6          #210 GPa = 210E6 kN/m2
48 Area  = 33.5E-4        #33.5 cm^2 = 33.5E-4 m^2
49 Inercia = 2140E-8      #2140 cm^4 = 2140E-8 m^4
50 Longitud = 2.8         #Longitud en metros
51
52 #=====
53 #Datos de desplazamientos conocidos en los extremos
54 #=====
55 #Datos del nudo inicial i
56 ui_x = 0
57 ui_y = 0
58 Theta_i = 0
59
60 #Datos del nudo final j
61 uj_x = 1E-3
62 uj_y = 1E-3
63 Theta_j = 1E-3
64
65 #=====
66 #Matriz de rigidez
67 #=====
68
69 EA = Elast * Area
70 EI = Elast * Inercia
71
72 Mrig = k_p2d(EA, EI, Longitud)
73
74 print("")
75 print("Matriz de Rigidez")
76 print(Mrig)

```

1. Conceptos básicos del Análisis Matricial

```
73 #Para que la salida por pantalla sea más clara,
74 #fijamos la salida con 3 cifras decimales
75 np.set_printoptions(precision=3)
76 print("")
77 print("Matriz de Rigidez")
78 print(Mrig)
79
80 #=====
81 #Vector desplazamientos
82 #=====
83 u = np.array([ui_x,ui_y,Theta_i,uj_x,uj_y,Theta_j])
84
85 #=====
86 #Vector de fuerzas
87 #=====
88 #Producto Matriz de rigidez * Vector de desplazamientos
89 #(observe que no es necesario escribir u como matriz-columna):
90 P_local = np.dot(Mrig,u)
91
92 print("")
93 print("Vector de fuerzas")
94 print(P_local)
95
96 #Asignación de las componentes del vector de fuerza:
97 Pi_x = P_local[0]
98 Pi_y = P_local[1]
99 M_i   = P_local[2]
100 Pj_x = P_local[3]
101 Pj_y = P_local[4]
102 Mj   = P_local[5]
103
104 #Salida por pantalla:
105 print("")
106 print("Pi_x = ", P_local[0])
107 print("Pi_y = ", P_local[1])
108 print("M_i   = ", P_local[2])
109 print("Pj_x = ", P_local[3])
110 print("Pj_y = ", P_local[4])
111 print("Mj   = ", P_local[5])
112
113 #=====
114 #Estudio paramétrico
115 #=====
116 Longitudes = np.linspace(0.5, 5, 50) #Vector desde 0.5 hasta 5,
    con 50 puntos intermedios
117 VecPi_y = []
118
119 for Longitud in Longitudes:
120     Mrigidez = k_p2d(EA, EI, Longitud)
121     P = np.dot(Mrigidez,u)
122     VecPi_y = np.append(VecPi_y,P[1]) #La posición [1] es la
    segunda, pues la primera es P[0]
123
124 print("")
125 print("Longitud,      Pi_y")
126 print("=====")
```

```

127 for i in range(len(Longitudes)):
128     print(Longitudes[i], ",", VecPi_y[i])
129
130
131 line, = plt.plot(Longitudes, VecPi_y, label = "P_{iy}",
132                 linewidth = 2.0)
133
134 plt.xlabel("Longitud de viga (m)")    #Etiqueta del eje x
135 plt.ylabel("kN")                      #Etiqueta del eje y
136
137 plt.legend(loc=0)    #Mostrar leyenda en posición óptima (loc=0)
138 plt.show()          #Mostrar figura final

```

Ejecutando el código, se obtiene como resultado que las fuerzas en el extremo libre de la viga son:

$$P_{jx} = 251.25 \text{ kN}; \quad P_{jy} = -0.98 \text{ kN}; \quad M_j = 2.98 \text{ kN} \cdot \text{m}$$

Y las reacciones en el empotramiento:

$$P_{ix} = -251.25 \text{ kN}; \quad P_{iy} = 0.98 \text{ kN}; \quad M_i = -0.23 \text{ kN} \cdot \text{m}$$

El estudio paramétrico muestra una relación no lineal entre la longitud de la viga y la reacción vertical en el empotramiento, lo cual se podía deducir observando que en la matriz de rigidez la longitud interviene en términos con potencias en los denominadores de las fracciones. Gráficamente se puede observar como un aumento de la longitud de la viga disminuye el módulo de la reacción vertical en el empotramiento a igualdad del resto de parámetros:

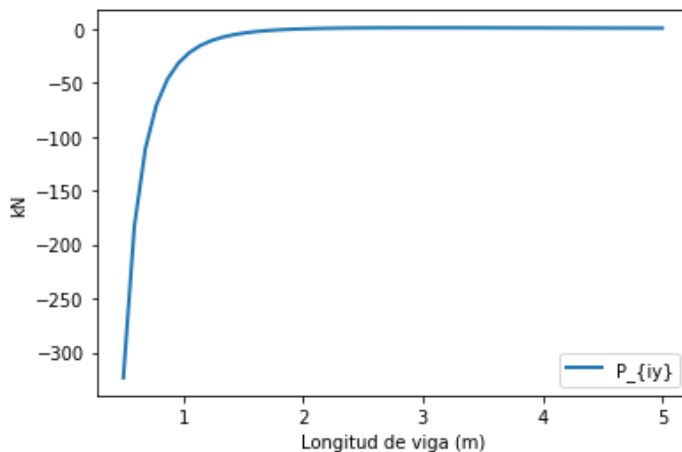


Figura 1.14: Ejercicio Python 1.1. Reacción vertical en el empotramiento en función de la longitud de la viga.

Capítulo 2

Sistemas de coordenadas y matrices de rigidez

Índice

2.1. Introducción	17
2.2. Sistemas de coordenadas	18
2.3. Matrices de rigidez elementales	24
2.4. Transformación de coordenadas	28

2.1. Introducción

En el Capítulo 1 se ha visto la ecuación matricial fundamental que se empleará para la resolución de problemas mediante el denominado *Método Directo de la Rigidez* (MDR). Esta ecuación relaciona el vector de fuerzas (\mathbf{F}) con el vector de desplazamientos (\mathbf{u}) a través de la matriz de rigidez de la estructura (\mathbf{K}):

$$\mathbf{F} = \mathbf{K}\mathbf{u}$$

Es posible discretizar esta expresión para cada una de las barras o elementos que componen la estructura, de la forma:

$$\mathbf{P}_L = \mathbf{k}_L \boldsymbol{\delta}_L \quad (2.1)$$

donde \mathbf{P}_L es el vector de fuerzas del elemento, \mathbf{k}_L es la matriz de rigidez del elemento y $\boldsymbol{\delta}_L$ el vector de desplazamientos del elemento. El subíndice “ L ” indica que el vector o matriz en cuestión está expresado en el sistema de coordenadas locales del elemento, como se describirá a continuación.

2.2. Sistemas de coordenadas

Como se ha introducido en el apartado anterior, las ecuaciones matriciales deben estar expresadas en un determinado sistema de coordenadas. Para la resolución de estructuras mediante el Cálculo Matricial necesitaremos cuatro sistemas de coordenadas diferentes: de referencia, global, local y nodal.

Sistema de coordenadas de referencia

Normalmente cartesiano, permite definir la geometría de la estructura y su posición en el espacio.

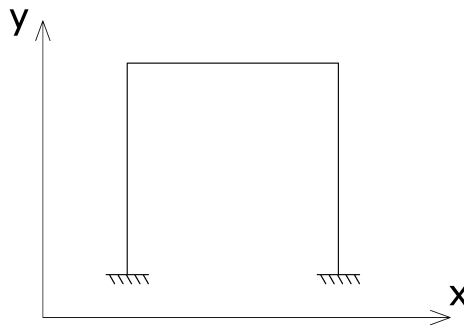


Figura 2.1: Sistema de coordenadas de referencia.

Sistema de coordenadas global

Permite definir de forma global y coherente en **toda la estructura** los **gdl**s de la misma, mediante su descripción en cada nudo. El número de **gdl** en cada nudo dependerán del tipo de estructura, de la forma en que se explica a continuación.

- **Estructura articulada plana (A2D):** 2 gdl por nudo.

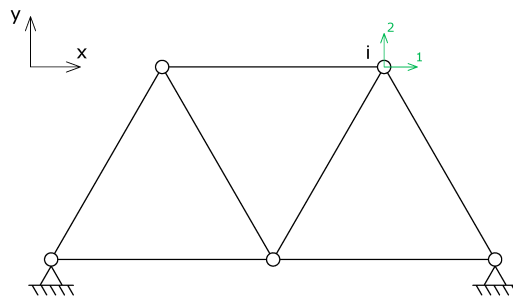


Figura 2.2: Estructura articulada plana.

Para un nudo cualquiera i :

$$\mathbf{F}_i = \begin{pmatrix} F_{ix} \\ F_{iy} \end{pmatrix} \quad \mathbf{u}_i = \begin{pmatrix} u_{ix} \\ u_{iy} \end{pmatrix}$$

- **Estructura articulada tridimensional (A3D):** 3 gdl por nudo.

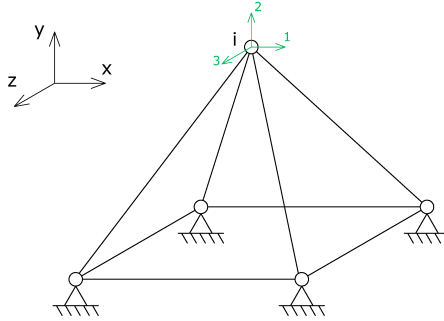


Figura 2.3: Estructura articulada tridimensional.

Para un nudo cualquiera i :

$$\mathbf{F}_i = \begin{pmatrix} F_{ix} \\ F_{iy} \\ F_{iz} \end{pmatrix} \quad \mathbf{u}_i = \begin{pmatrix} u_{ix} \\ u_{iy} \\ u_{iz} \end{pmatrix}$$

- **Pórtico plano (P2D):** 3 gdl por nudo.

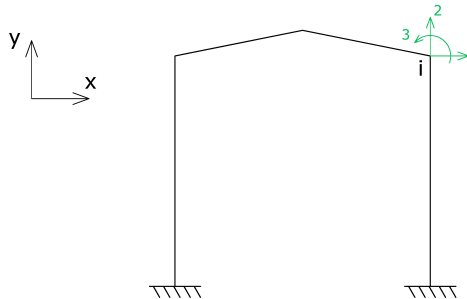


Figura 2.4: Pórtico plano.

Para un nudo cualquiera i :

$$\mathbf{F}_i = \begin{pmatrix} F_{ix} \\ F_{iy} \\ M_i \end{pmatrix} \quad \mathbf{u}_i = \begin{pmatrix} u_{ix} \\ u_{iy} \\ \theta_i \end{pmatrix}$$

- **Pórtico tridimensional (P3D):** 6 gdl por nudo.

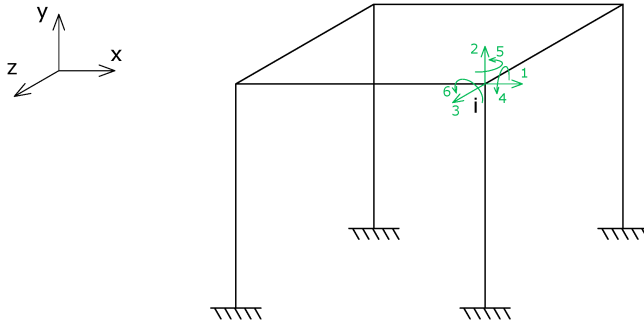


Figura 2.5: Pórtico tridimensional.

Para un nudo cualquiera i :

$$\mathbf{F}_i = \begin{pmatrix} F_{ix} \\ F_{iy} \\ F_{iz} \\ M_{ix} \\ M_{iy} \\ M_{iz} \end{pmatrix} \quad \mathbf{u}_i = \begin{pmatrix} u_{ix} \\ u_{iy} \\ u_{iz} \\ \theta_{ix} \\ \theta_{iy} \\ \theta_{iz} \end{pmatrix}$$

- **Emparrillado plano (EMP):** 3 gdl por nudo.

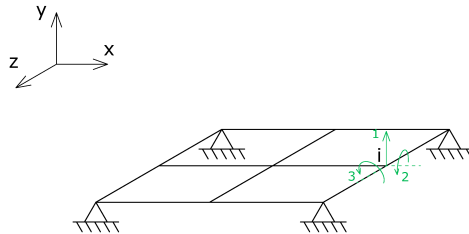


Figura 2.6: Emparrillado

Para un nudo cualquiera i :

$$\mathbf{F}_i = \begin{pmatrix} F_{iy} \\ M_{iz} \\ M_{ix} \end{pmatrix} \quad \mathbf{u}_i = \begin{pmatrix} u_{iy} \\ \theta_{iz} \\ \theta_{ix} \end{pmatrix}$$

Sistema de coordenadas local

Es un sistema de coordenadas asociado a **cada elemento**, permitiendo definir las ecuaciones matriciales en todos ellos de la misma forma, independientemente de su posición en el espacio.

Los sistemas de coordenadas locales según el tipo de estructura son:

- **Estructura articulada plana (A2D):** 2 gdl por elemento.

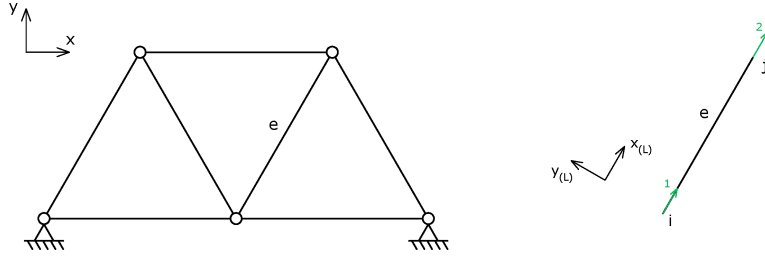


Figura 2.7: Elemento en estructura articulada plana.

Para un elemento cualquiera e de extremos $i - j$:

$$\mathbf{P}_L = \begin{pmatrix} P_{ix,L} \\ P_{jx,L} \end{pmatrix} \quad \boldsymbol{\delta}_L = \begin{pmatrix} \delta_{ix,L} \\ \delta_{jx,L} \end{pmatrix}$$

- **Estructura articulada tridimensional (A3D):** 2 gdl por elemento.

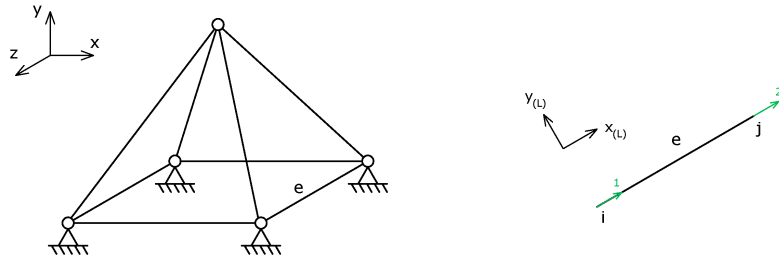


Figura 2.8: Elemento en estructura articulada tridimensional.

Para un elemento cualquiera e de extremos $i - j$:

$$\mathbf{P}_L = \begin{pmatrix} P_{ix,L} \\ - \\ P_{jx,L} \end{pmatrix} \quad \boldsymbol{\delta}_L = \begin{pmatrix} \delta_{ix,L} \\ - \\ \delta_{jx,L} \end{pmatrix}$$

- **Pórtico plano (P2D):** 6 gdl por elemento.

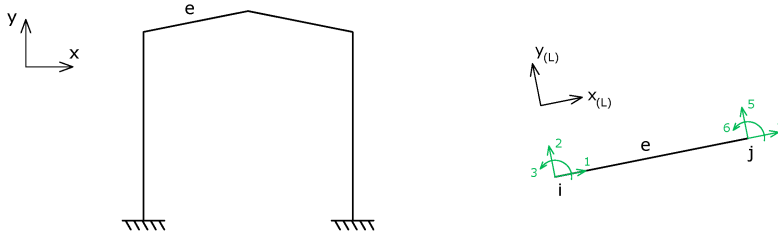


Figura 2.9: Elemento en pórtico plano.

Para un elemento cualquiera e de extremos $i - j$:

$$\mathbf{P}_L = \begin{pmatrix} P_{ix,L} \\ P_{iy,L} \\ M_{i,L} \\ - \\ P_{jx,L} \\ P_{jy,L} \\ M_{j,L} \end{pmatrix} \quad \boldsymbol{\delta}_L = \begin{pmatrix} \delta_{ix,L} \\ \delta_{iy,L} \\ \theta_{i,L} \\ - \\ \delta_{jx,L} \\ \delta_{jy,L} \\ \theta_{j,L} \end{pmatrix}$$

- **Pórtico tridimensional (P3D):** 12 gdl por elemento.

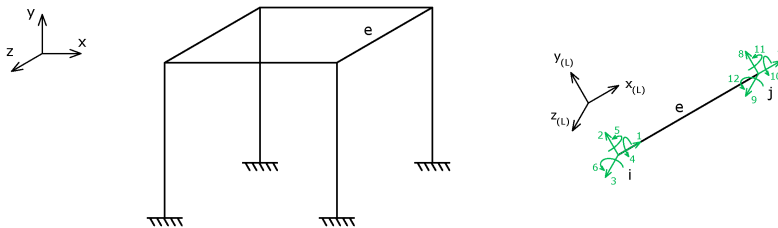


Figura 2.10: Elemento en pórtico tridimensional.

Para un elemento cualquiera e de extremos $i - j$:

$$\mathbf{P}_L = \begin{pmatrix} P_{ix,L} \\ P_{iy,L} \\ P_{iz,L} \\ M_{ix,L} \\ M_{iy,L} \\ M_{iz,L} \\ P_{jx,L} \\ P_{jy,L} \\ P_{jz,L} \\ M_{jx,L} \\ M_{jy,L} \\ M_{jz,L} \end{pmatrix} \quad \boldsymbol{\delta}_L = \begin{pmatrix} \delta_{ix,L} \\ \delta_{iy,L} \\ \delta_{iz,L} \\ \theta_{ix,L} \\ \theta_{iy,L} \\ \theta_{iz,L} \\ \delta_{jx,L} \\ \delta_{jy,L} \\ \delta_{jz,L} \\ \theta_{jx,L} \\ \theta_{jy,L} \\ \theta_{jz,L} \end{pmatrix}$$

- **Emparrillado (EMP):** 6 gdl por elemento.

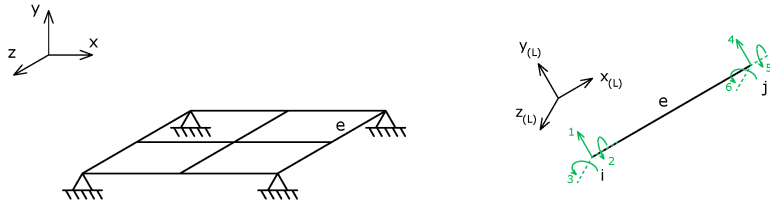


Figura 2.11: Elemento en emparrillado.

Para un elemento cualquiera e de extremos $i - j$:

$$\mathbf{P}_L = \begin{pmatrix} P_{iy,L} \\ M_{iz,L} \\ M_{ix,L} \\ P_{jy,L} \\ M_{jz,L} \\ M_{jx,L} \end{pmatrix} \quad \boldsymbol{\delta}_L = \begin{pmatrix} \delta_{iy,L} \\ \theta_{iz,L} \\ \theta_{ix,L} \\ \delta_{jy,L} \\ \theta_{jz,L} \\ \theta_{jx,L} \end{pmatrix}$$

Sistema de coordenadas nodal

Si las condiciones de contorno de un determinado **nudo** no se ajustan al sistema de coordenadas global, resulta útil definir un sistema de coordenadas particular para dicho nudo. De forma análoga a la nomenclatura utilizada para el sistema local, el uso de un sistema de coordenadas nodal se indicará con el subíndice “ N ”.

En la siguiente figura se muestra el ejemplo de un apoyo con deslizadera no concordante con el sistema global de coordenadas, donde es necesario definir un sistema de coordenadas nodal.

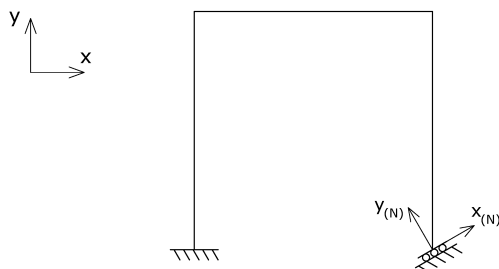


Figura 2.12: Sistema de coordenadas nodal en apoyo no concordante.

La utilidad del uso de un sistema de coordenadas nodal para apoyos no concordantes se verá con profundidad en el Capítulo 5.

2.3. Matrices de rigidez elementales

Según la definición de matriz de rigidez, cada elemento K_{ij} de la misma representa la fuerza según la coordenada i que provoca un movimiento unitario según la coordenada j , manteniendo todos los demás iguales a cero. De este modo, la matriz de rigidez de cada elemento puede ser generada por columnas aplicando desplazamientos unitarios en cada **gdl**. Este cálculo se realiza en coordenadas locales del elemento, para que la matriz obtenida sea independiente de la posición del mismo.

A continuación se muestran las matrices de rigidez elementales para los tipos estructurales más habituales.

Elemento en estructura articulada 2D o 3D

Como se ha indicado, la matriz de rigidez se puede obtener por columnas aplicando desplazamientos unitarios para cada **gdl**. De este modo, la primera columna de la matriz se obtiene aplicando un desplazamiento unitario en el **gdl** 1:

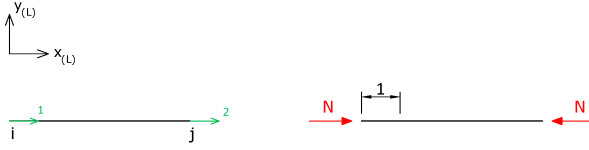


Figura 2.13: Elemento en A2D o A3D, desplazamiento unitario en el gdl 1.

Aplicando la Ley de Hooke, es fácil obtener el valor de la fuerza \mathcal{N} :

$$\sigma = E\varepsilon \Rightarrow \frac{\mathcal{N}}{A} = E \frac{\Delta L}{L} \Rightarrow \mathcal{N} = \frac{EA}{L} \cdot 1 = \frac{EA}{L}$$

donde E es el módulo de elasticidad de la barra y A el área de su sección.

Para la segunda columna, aplicando un desplazamiento unitario en el gdl 2:

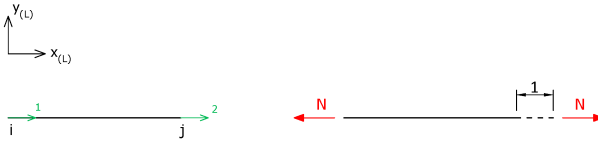


Figura 2.14: Elemento en A2D o A3D, desplazamiento unitario en el gdl 2.

Siendo, igualmente, $\mathcal{N} = EA/L$.

Por tanto, la matriz de rigidez elemental para una barra de estructura articulada queda:

$$\mathbf{k}_L = \begin{pmatrix} \frac{EA}{L} & 0 \\ 0 & \frac{EA}{L} \end{pmatrix} \quad (2.2)$$

Elemento en pórtico 2D

El procedimiento es igual al indicado para el caso de la estructura articulada. Gráficamente, para las tres primeras columnas (siendo E el módulo de elasticidad de la barra y A e I el área y el momento de inercia de su sección, respectivamente):

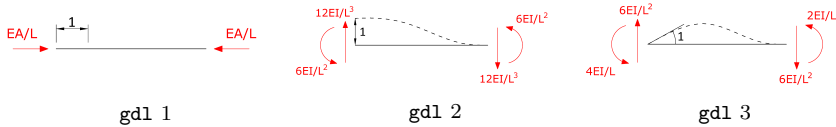


Figura 2.15: Elemento en P2D, desplazamientos unitarios en los 3 primeros gdl.

Por simetría es posible obtener los elementos correspondientes a los desplazamientos unitarios en el extremo j (columnas 4, 5 y 6 de la matriz), resultando la matriz de rigidez elemental siguiente:

$$\mathbf{k}_L = \left(\begin{array}{ccc|ccc} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ \hline -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{array} \right)$$

Elemento en pórtico 3D

Siguiendo el mismo procedimiento, para un elemento de pórtico tridimensional, con 6 gdl por nudo (12 por elemento), se obtiene la siguiente matriz 12×12 :

$$\mathbf{k}_L = \left(\begin{array}{c|c} \mathbf{k}_{ii,L} & \mathbf{k}_{ij,L} \\ \hline \mathbf{k}_{ji,L} & \mathbf{k}_{jj,L} \end{array} \right)$$

donde

$$\mathbf{k}_{ii,L} = \left(\begin{array}{cccccc} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\ 0 & 0 & \frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\ 0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} \end{array} \right)$$

$$\mathbf{k}_{ij,L} = \left(\begin{array}{cccccc} -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\ 0 & 0 & -\frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 \\ 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\ 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \end{array} \right)$$

$$\mathbf{k}_{ji,L} = \begin{pmatrix} -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \\ 0 & 0 & -\frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\ 0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \end{pmatrix}$$

$$\mathbf{k}_{jj,L} = \begin{pmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \\ 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\ 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\ 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} \end{pmatrix}$$

donde, además de los parámetros ya descritos, intervienen el módulo de elasticidad transversal o módulo de cizalladura (G) y el módulo de torsión (J). Además, en este caso es necesario distinguir entre la inercia de la sección según el eje y y el eje z del elemento (I_y e I_z , respectivamente).

Elemento en emparrillado plano

Por último, para un elemento de emparrillado:

$$\mathbf{k}_L = \begin{pmatrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI_z}{L^2} & 0 \\ \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} & 0 \\ 0 & 0 & \frac{GJ}{L} & 0 & 0 & -\frac{GJ}{L} \\ -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 \\ \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} & 0 \\ 0 & 0 & -\frac{GJ}{L} & 0 & 0 & \frac{GJ}{L} \end{pmatrix}$$

Los elementos que se han presentado son los más usuales, pero podrían definirse matrices elementales para otros tipos de elementos según las necesidades. En el Capítulo 5 se incluyen algunos de ellos.

Es importante, además, tener en cuenta que en los cálculos realizados no se ha tenido en cuenta la deformación por cortante de las barras.

2.4. Transformación de coordenadas

Para el ensamblaje de las ecuaciones de los elementos, a fin de obtener la ecuación de rigidez de la estructura, es necesario transformar las matrices y vectores de un sistema de coordenadas a otro. A continuación se describen los cambios de coordenadas del sistema nodal al global y del sistema local al global y viceversa, para estructuras de tipo A2D y P2D. El procedimiento para obtener el cambio de coordenadas en las otras tipologías estructurales es análogo.

Cambio nodal-global en estructura A2D

Supongamos en un nudo cualquiera de una estructura dos pares de ejes cartesianos, uno global y otro nodal, formando entre sí un ángulo α . Supongamos también un vector de fuerzas $\mathbf{P}_N = \{P_{x,N}, P_{y,N}\}$ descompuesto según los ejes nodales:

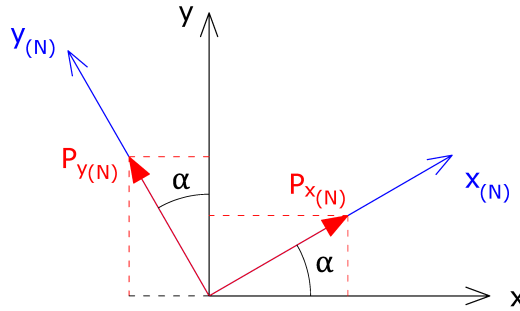


Figura 2.16: Transformación de coordenadas nodal-global.

Las componentes de dicho vector \mathbf{P} en coordenadas globales serán: ¹

$$P_{x,G} = P_{x,N} \cos \alpha - P_{y,N} \sin \alpha$$

$$P_{y,G} = P_{x,N} \sin \alpha + P_{y,N} \cos \alpha$$

Por simplicidad, en adelante escribiremos c y s como $\cos \alpha$ y $\sin \alpha$, respectivamente. De este modo, y expresándolo en forma matricial, tenemos:

$$\begin{pmatrix} P_{x,G} \\ P_{y,G} \end{pmatrix} = \begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} P_{x,N} \\ P_{y,N} \end{pmatrix} \Rightarrow \mathbf{P}_G = \mathbf{m}_{NG} \mathbf{P}_N$$

¹Para facilitar la comprensión, en el presente apartado se incluye el subíndice “G” a los elementos en coordenadas globales. De forma general y por simplicidad, este subíndice se omitirá, quedando sin subíndice los vectores y matrices que se expresen coordenadas globales.

donde se denomina \mathbf{m}_{NG} a la **matriz de cambio de coordenadas de nodales a globales**, que en el caso de una estructura A2D es de la forma:

$$\mathbf{m}_{NG} = \begin{pmatrix} c & -s \\ s & c \end{pmatrix}$$

Es importante tener en cuenta que el ángulo α utilizado para definir la matriz de cambio se mide desde el eje global hacia el nodal en sentido positivo (antihorario).

Evidentemente, este cambio no sólo es aplicable al vector de fuerzas, sino a cualquier vector en el sistema. En el caso del vector de desplazamientos:

$$\delta_G = \mathbf{m}_{NG} \delta_N$$

Si atendemos ahora a la transformación inversa, de coordenadas globales a nodales, tenemos:

$$\begin{aligned} P_{x,N} &= P_{x,G} \cos \alpha + P_{y,G} \sin \alpha \\ P_{y,N} &= -P_{x,G} \sin \alpha + P_{y,G} \cos \alpha \end{aligned}$$

En forma matricial:

$$\begin{pmatrix} P_{x,N} \\ P_{y,N} \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} P_{x,G} \\ P_{y,G} \end{pmatrix} \Rightarrow \mathbf{P}_N = \mathbf{m}_{GN} \mathbf{P}_G$$

donde se observa que la matriz de cambio de coordenadas nodales a globales es la traspuesta de la matriz de cambio de coordenadas globales a nodales, y coincide en este caso con la inversa:

$$\mathbf{m}_{NG} = \mathbf{m}_{GN}^{-1} = \mathbf{m}_{GN}^T$$

Como se ha indicado, los cambios de coordenadas son aplicables a cualquier vector. Para el vector desplazamientos:

$$\delta_N = \mathbf{m}_{GN} \delta_G$$

Cambio nodal-global en estructura P2D

Tomamos como ejemplo de nuevo un vector de fuerzas \mathbf{P} . Al tratarse de una estructura tipo pórtico 2D, este vector tendrá una tercera componente correspondiente al momento, la cual no varía al girar el sistema de coordenadas:

$$\begin{aligned} P_{x,G} &= P_{x,N} \cos \alpha - P_{y,N} \sin \alpha \\ P_{y,G} &= P_{x,N} \sin \alpha + P_{y,N} \cos \alpha \\ M_G &= M_N \end{aligned}$$

En forma matricial:

$$\begin{pmatrix} P_{x,G} \\ P_{y,G} \\ M_G \end{pmatrix} = \begin{pmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_{x,N} \\ P_{y,N} \\ M_N \end{pmatrix} \Rightarrow \mathbf{P}_G = \mathbf{m}_{NG} \mathbf{P}_N$$

siendo entonces la **matriz de cambio de coordenadas nodales a globales** para una estructura P2D:

$$\mathbf{m}_{NG} = \begin{pmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

La transformación inversa, donde se cumple de nuevo que $\mathbf{m}_{NG} = \mathbf{m}_{GN}^T$, será:

$$\begin{pmatrix} P_{x,N} \\ P_{y,N} \\ M_N \end{pmatrix} = \begin{pmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_{x,G} \\ P_{y,G} \\ M_G \end{pmatrix} \Rightarrow \mathbf{P}_N = \mathbf{m}_{GN} \mathbf{P}_G$$

Al igual que en el caso anterior, el cambio es válido para cualquier vector, incluido el vector de desplazamientos.

Cambio local-global en estructura A2D

Tomemos de nuevo el ejemplo del vector fuerzas para obtener el cambio de coordenadas de locales a globales. Las barras de una estructura articulada sólo trabajan a axil (compresión/tracción), por lo que en cada nudo el vector fuerza sólo tiene componentes en la dirección x local de la barra. Al descomponer estas fuerzas según los ejes globales, se obtendrá una componente en la dirección x y otra en la dirección y del sistema global.

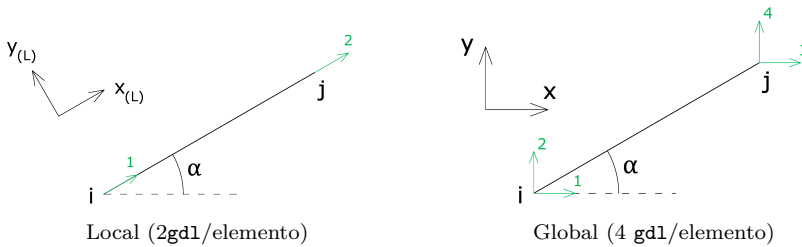


Figura 2.17: Transformación de coordenadas local-global para A2D.

Esto implica que el vector de fuerzas \mathbf{P} tendrá dimensiones 2×1 en coordenadas locales (una fuerza por nodo en dirección x_L) y sin embargo tendrá dimensiones 4×1 en coordenadas globales (dos componentes, x e y , para la fuerzas en cada nodo):

$$\mathbf{P}_L = \begin{pmatrix} P_{ix,L} \\ P_{jx,L} \end{pmatrix}; \quad \mathbf{P}_G = \begin{pmatrix} P_{i,G} \\ P_{j,G} \end{pmatrix} = \begin{pmatrix} P_{ix,G} \\ P_{iy,G} \\ P_{jx,G} \\ P_{jy,G} \end{pmatrix}$$

El cambio de locales a globales será:

$$P_{ix,G} = P_{ix,L} \cos \alpha$$

$$P_{iy,G} = P_{ix,L} \sin \alpha$$

$$P_{jx,G} = P_{jx,L} \cos \alpha$$

$$P_{jy,G} = P_{jx,L} \sin \alpha$$

que matricialmente queda:

$$\begin{pmatrix} P_{ix,G} \\ P_{iy,G} \\ P_{jx,G} \\ P_{jy,G} \end{pmatrix} = \begin{pmatrix} c & 0 \\ s & 0 \\ 0 & c \\ 0 & s \end{pmatrix} \begin{pmatrix} P_{ix,L} \\ P_{jx,L} \end{pmatrix} \Rightarrow \mathbf{P}_G = \mathbf{M}_{LG} \mathbf{P}_L \quad (2.3)$$

Donde \mathbf{M}_{LG} es la **matriz de cambio de coordenadas de locales a globales**. La expresión 2.3 se puede expresar también como:

$$\begin{pmatrix} P_{i,G} \\ P_{j,G} \end{pmatrix} = \begin{pmatrix} \mathbf{m}_{LG} & \mathbf{0} \\ \mathbf{0} & \mathbf{m}_{LG} \end{pmatrix} \begin{pmatrix} P_{i,L} \\ P_{j,L} \end{pmatrix}$$

siendo

$$\mathbf{m}_{LG} = \begin{pmatrix} c \\ s \end{pmatrix}$$

la matriz de transformación de locales a globales para cada uno de los nudos de la barra.

El cambio inverso, de coordenadas globales a locales, queda de la siguiente forma:

$$\begin{pmatrix} P_{ix,L} \\ P_{jx,L} \end{pmatrix} = \begin{pmatrix} c & s & 0 & 0 \\ 0 & 0 & c & s \end{pmatrix} \begin{pmatrix} P_{ix,G} \\ P_{iy,G} \\ P_{jx,G} \\ P_{jy,G} \end{pmatrix} \Rightarrow$$

$$\Rightarrow \begin{pmatrix} P_{i,L} \\ P_{j,L} \end{pmatrix} = \begin{pmatrix} m_{GL} & 0 \\ 0 & m_{GL} \end{pmatrix} \begin{pmatrix} P_{i,G} \\ P_{j,G} \end{pmatrix}$$

De nuevo $M_{LG} = M_{GL}^T$ y también $m_{LG} = m_{GL}^T$.

Cambio local-global en estructura P2D

En este caso tenemos 6 gdl/elemento tanto en el sistema local como en el global.

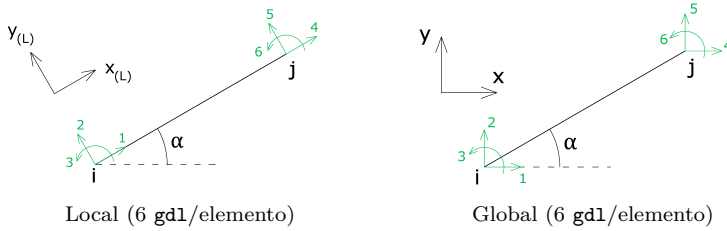


Figura 2.18: Transformación de coordenadas local-global para P2D.

Las componentes del vector fuerza en el nodo i (análogo para j) en coordenadas globales son:

$$\begin{aligned} P_{ix,G} &= P_{ix,L} \cos \alpha - P_{iy,L} \sin \alpha \\ P_{iy,G} &= P_{ix,L} \sin \alpha + P_{iy,L} \cos \alpha \\ M_{i,G} &= M_{i,L} \end{aligned}$$

Expresado matricialmente para el elemento de nodos $i - j$:

$$\begin{pmatrix} P_{ix,G} \\ P_{iy,G} \\ M_{i,G} \\ P_{jx,G} \\ P_{jy,G} \\ M_{j,G} \end{pmatrix} = \begin{pmatrix} c & -s & 0 & 0 & 0 & 0 \\ s & c & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & -s & 0 \\ 0 & 0 & 0 & s & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_{ix,L} \\ P_{iy,L} \\ M_{i,L} \\ P_{jx,L} \\ P_{jy,L} \\ M_{j,L} \end{pmatrix} \Rightarrow \mathbf{P}_G = \mathbf{M}_{LG} \mathbf{P}_L$$

Que, igual que en el caso anterior, se puede expresar por nodos:

$$\begin{pmatrix} \mathbf{P}_{i,G} \\ \mathbf{P}_{j,G} \end{pmatrix} = \begin{pmatrix} \mathbf{m}_{LG} & \mathbf{0} \\ \mathbf{0} & \mathbf{m}_{LG} \end{pmatrix} \begin{pmatrix} \mathbf{P}_{i,L} \\ \mathbf{P}_{j,L} \end{pmatrix}$$

La transformación inversa (global-local) se realiza mediante la matriz $\mathbf{M}_{GL} = \mathbf{M}_{LG}^T$; cumpliéndose también que $\mathbf{m}_{LG} = \mathbf{m}_{GL}^T$.

A continuación se muestra un ejemplo sencillo de aplicación de los cambios de coordenadas de vectores para hallar los esfuerzos en una barra.

Ejercicio 2.1 Calcular los esfuerzos en la barra de la figura, perteneciente a un pórtico 2D, de la que se conocen los desplazamientos en sus extremos según los ejes cartesianos globales XY :

$$\delta_{1x} = \delta_{1y} = \theta_1 = 0 \text{ (extremo 1 empotrado)}$$

$$\delta_{2x} = 3.088 \cdot 10^{-2} \text{ m}; \quad \delta_{2y} = -3.100 \cdot 10^{-2} \text{ m}; \quad \theta_2 = -1.317 \cdot 10^{-3} \text{ rad}$$

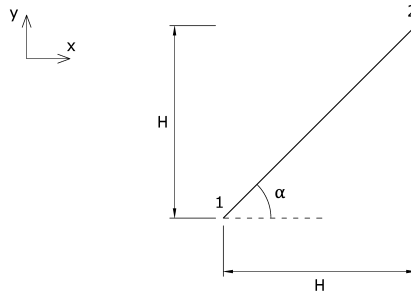


Figura 2.19: Ejercicio 2.1. Elemento P2D.

Datos: $H = 5 \text{ m}$; $E = 2 \cdot 10^6 \text{ t/m}^2$; $I = 1.8 \cdot 10^{-3} \text{ m}^4$; $A = 0.2 \text{ m}^2$.

Solución

Los esfuerzos se pueden obtener directamente a través de las fuerzas en coordenadas locales de la barra. Estas se obtienen, aplicando la ecuación 2.1, como:

$$\mathbf{P}_L = \mathbf{k}_L \boldsymbol{\delta}_L$$

La matriz de rigidez \mathbf{k}_L es la matriz elemental para un elemento P2D, mostrada en el apartado 2.3. Para poder aplicarla necesitamos transformar los desplazamiento a coordenadas locales, que aplicando las ecuaciones de transformación vistas:

$$\boldsymbol{\delta}_L = \mathbf{M}_{GL} \boldsymbol{\delta}$$

Al tratarse de una barra P2D:

$$\begin{pmatrix} \delta_{1x,L} \\ \delta_{1y,L} \\ \theta_{1,L} \\ \delta_{2x,L} \\ \delta_{2y,L} \\ \theta_{2,L} \end{pmatrix} = \begin{pmatrix} c & s & 0 & 0 & 0 & 0 \\ -s & c & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & s & 0 \\ 0 & 0 & 0 & -s & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \delta_{1x} \\ \delta_{1y} \\ \theta_1 \\ \delta_{2x} \\ \delta_{2y} \\ \theta_2 \end{pmatrix}$$

donde $c = \cos \alpha$, $s = \sin \alpha$ y $\alpha = 45^\circ$. Sustituyendo y operando, se obtiene:

$$\boldsymbol{\delta}_L = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -8.485 \cdot 10^{-5} \\ -4.376 \cdot 10^{-2} \\ -1.317 \cdot 10^{-3} \end{pmatrix}$$

De esta forma, el vector de fuerzas en coordenadas locales es:

$$\begin{pmatrix} P_{1x,L} \\ P_{1y,L} \\ M_{1,L} \\ P_{2x,L} \\ P_{2y,L} \\ M_{2,L} \end{pmatrix} = \begin{pmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^3} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^3} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^3} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^3} & \frac{4EI}{L} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ -8.485 \cdot 10^{-5} \\ -4.376 \cdot 10^{-2} \\ -1.317 \cdot 10^{-3} \end{pmatrix}$$

que, sustituyendo los valores de los parámetros y operando, queda:

$$P_L = \begin{pmatrix} 4.80 \\ 4.78 \\ 17.56 \\ -4.80 \\ -4.78 \\ 16.22 \end{pmatrix}$$

Gráficamente:

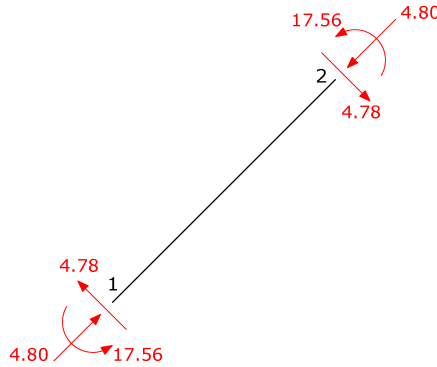


Figura 2.20: Ejercicio 2.1. Fuerzas en la barra.

Se observa que la barra está en equilibrio. Representamos por último las leyes de esfuerzos en el elemento:

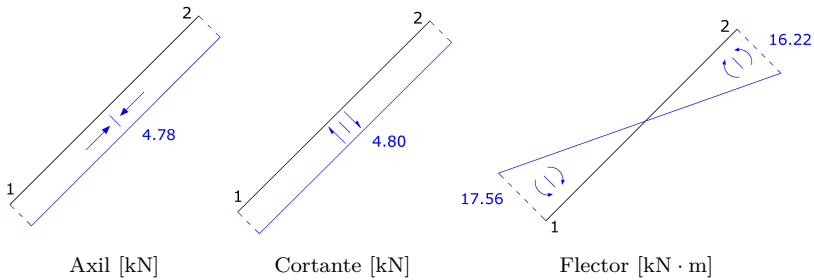


Figura 2.21: Ejercicio 2.1. Esfuerzos en la barra.

Cambio de coordenadas de la matriz de rigidez

Hasta el momento hemos visto cómo cambiar de coordenadas los vectores que intervienen en el problema, veamos ahora cómo hacerlo para la matriz de rigidez.

Comenzamos con el caso de cambio de locales a globales. De lo explicado anteriormente, sabemos que:

$$\left. \begin{array}{l} P_L = k_L \delta_L \\ \delta_L = M_{GL} \delta_G \end{array} \right\} \Rightarrow P_L = k_L M_{GL} \delta_G$$

Multiplicando por M_{LG} por la izquierda:

$$M_{LG} P_L = M_{LG} k_L M_{GL} \delta_G$$

Como sabemos que $M_{LG} P_L = P_G$, entonces:

$$P_G = M_{LG} k_L M_{GL} \delta_G$$

Y dado que se debe verificar que $P_G = k_G \delta_G$, podemos deducir que:

$$k_G = M_{LG} k_L M_{GL}$$

La expresión obtenida es aplicable análogamente al cambio inverso, de globales a locales, y también a los cambios de coordenadas entre los sistemas nodal y global.

De forma general, podemos decir que para transformar la matriz de rigidez de un sistema de coordenadas A a un sistema B , se debe multiplicar por la izquierda por la matriz de cambio de A a B y por la derecha por la matriz de cambio de B a A :

$$k_B = M_{AB} k_A M_{BA}$$

Ejercicio 2.2 *Obtener la matriz de rigidez de una barra de estructura articulada bidimensional en coordenadas globales. La barra se encuentra formando un ángulo α con el eje global X .*

Solución

Conocemos la matriz de rigidez en coordenadas locales de un elemento A2D (Ecuación 2.2) y la matriz de cambio de coordenadas locales a globales para estructuras A2D (Ecuación 2.3), así como su transpuesta. De esta forma, aplicando la expresión que permite la transformación de coordenadas de la

matriz de rigidez:

$$\mathbf{k}_G = \mathbf{M}_{LG} \mathbf{k}_L \mathbf{M}_{GL} = \begin{pmatrix} c & 0 \\ s & 0 \\ 0 & c \\ 0 & s \end{pmatrix} \begin{pmatrix} \frac{EA}{L} & -\frac{EA}{L} \\ -\frac{EA}{L} & \frac{EA}{L} \end{pmatrix} \begin{pmatrix} c & s & 0 & 0 \\ 0 & 0 & c & s \end{pmatrix}$$

donde, recordemos, llamamos c y s al coseno y al seno del ángulo α , respectivamente. Operando:

$$\mathbf{k}_G = \frac{EA}{L} \begin{pmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{pmatrix}$$

Se puede observar que la matriz pasa de tener dimensiones 2×2 en coordenadas locales (2 gdl/elemento) a tener dimensiones 4×4 en coordenadas globales (4 gdl/elemento).

Ejercicio Python 2.1

Sea una barra de tipo A3D que va desde el nudo $i = (10, 10, 0)$ al $j = (20, 30, 60)$ (coordenadas en metros). La barra tiene módulo de elasticidad igual a 210 GPa y una sección de área 2 cm^2 . Calculemos la matriz de rigidez de la barra en coordenadas locales y globales.

El procedimiento seguido en este caso es el siguiente:

1. Crear una clase “Punto” que genera puntos a partir de coordenadas, así como una función capaz de calcular la distancia entre dos puntos (longitud de la barra).
2. Crear una clase “A3D” para representar barras A3D a partir de un punto inicial, un punto final y unas propiedades EA . La clase “A3D” incluye una función para calcular la matriz de rigidez de la barra A3D en locales, otra para calcular la matriz de cambio de coordenadas de locales a globales y otra que usa las dos anteriores para obtener la matriz de rigidez en coordenadas globales.
3. Se introducen los datos del problema utilizando la clase “Punto”.
4. Se crea la barra mediante la clase “A3D”.
5. Se utilizan las funciones de la clase “A3D” para obtener las matrices de rigidez en coordenadas locales, de cambio local-global y de rigidez en coordenadas globales.

6. Se imprimen los resultados.

```
1 #=====
2 #Librerías y funciones necesarias
3 #=====
4 from __future__ import division
5 import numpy as np
6
7 #=====
8 #Clases y funciones para matriz de rigidez A3D
9 #Programación orientada a objetos
10 #=====
11
12 #-----
13 #Definición de la clase "Punto". Coordenadas (x, y, z) de un
14 # punto
15 # Observación: Se inicializa con las coordenadas (0,0,0)
16 # Ejemplo: p1 = Punto() generará un punto p1 de coordenadas
17 # (0,0,0)
18 # p1 = Punto(1,2,3) generará un punto de coordenadas
19 # (1,2,3)
20 # Uso de la función coords()
21 # Muestra por pantalla el punto, en formato (x,y,z)
22 # Ejemplo: p1 = Punto(1,2,3)
23 # p1.coords()
24 # (Se muestra en pantalla (1,2,3) )
25 #-----
26
27 class Punto:
28     """ Clase para representar los puntos en 3D, coordenadas x,
29     y, z """
30     def __init__(self, x=0, y=0, z=0):
31         self.x = x
32         self.y = y
33         self.z = z
34     def coords(self):
35         return "{0}, {1},{2}".format(self.x, self.y, self.z)
36
37 #-----
38 #Definición de la función "longitud"
39 # Esta función calcula la distancia entre dos puntos
40 # Notas : += es una función interesante, heredada de C
41 # Ejemplo: a = a + b se escribe a += b
42 # Esta función trabaja con p1,p2. Es una función "amiga" de
43 # la clase Punto
44 #-----
45
46 def longitud(p1, p2):
47     """ Devuelve la longitud (módulo) del vector p1-p2 """
48     Longitud = 0
49     Longitud += (p2.x - p1.x ) ** 2
50     Longitud += (p2.y - p1.y ) ** 2
51     Longitud += (p2.z - p1.z ) ** 2
52     Longitud = np.sqrt(Longitud)
53     return Longitud
54
55 #-----
```

```

50 # Definición de la clase A3D, para barra articulada 3D
51 #-----
52 class A3D:
53     """ Clase para representar la barra articulada 3D """
54     def __init__(self, p1 = Punto(), p2 = Punto(), EA = 0):
55         self.p1 = p1
56         self.p2 = p2
57         self.ea = EA
58
59 #Función para obtener la matriz de rigidez A3D en locales (2x2)
60     def mrig_local(self):
61         L = longitud(self.p1,self.p2)
62         K_L = np.array([[self.ea / L, -self.ea / L],
63                        [-self.ea / L, self.ea / L]])
64         return K_L
65
66 #Función para obtener la matriz de cambio de base de Local a
67   Global
68 #Para matrices A3D
69     def m_lg(self):
70         L = longitud (self.p1, self.p2) #Se usa la función
71         definida antes "longitud"
72         #Definición del vector unitario de dirección p1
73         apuntando a p2
74         vx = (self.p2.x - self.p1.x) / L
75         vy = (self.p2.y - self.p1.y) / L
76         vz = (self.p2.z - self.p1.z) / L
77         #Montaje de la matriz de cambio de base de Local a
78         Global.
79         #Recuerde: Por columnas, se indican los vectores del
80         sistema Local, referenciados en el sistema Global.
81         M_LG = np.array([[vx, 0],
82                        [vy, 0],
83                        [vz, 0],
84                        [0, vx],
85                        [0, vy],
86                        [0, vz]])
87         return M_LG
88
89 #Función para obtener la matriz de rigidez A3D en globales (6x6)
90     def mrig_global(self):
91
92         M_LG = self.m_lg() # Se llama a la función previamente
93         creada
94         K_L = self.mrig_local() # Se llama a otra función
95         previamente creada
96         #Producto matricial para construir la matriz de rigidez
97         en globales.
98         K_G = np.dot(M_LG, np.dot(K_L , M_LG.T ))
99
100        return K_G
101
102 #Función para escribir por pantalla la matriz de rigidez A3D
103     def imprime_mrig(self):
104         np.set_printoptions(precision=3)
105         print(self.mrig_global())

```

2. Sistemas de coordenadas y matrices de rigidez

```
98
99
100 #=====
101 # Datos de entrada, usamos la clase "Punto" creada
102 #=====
103 Punto1 = Punto(10,10,0)
104 Punto2 = Punto(20,30,60)
105 Elast = 210E6 # kN/m2
106 Area = 2E-4 # 2 cm2, pasado a m2
107
108 EA = Elast * Area
109
110 #=====
111 # Creación de un objeto de tipo A3D, a través de la clase A3D
112 #=====
113 barraA = A3D(Punto1, Punto2, EA)
114
115 #=====
116 # Generación de las matrices de rigidez y de cambio
117 #=====
118 # Usaremos las funciones del objeto "barraA" de clase A3D.
119 # Para usar una función de la clase A3D, se escribe
120     nombredeobjeto.función(argumentos)
121
122 matrig = barraA.mrig_global() # matriz de rigidez local
123 mat_lg = barraA.m_lg() # matriz cambio local-global
124 matrig_local = barraA.mrig_local() # matriz de rigidez global
125
126 #=====
127 # Impresión de resultados
128 #=====
129 print("Matriz de rigidez en coord. locales")
130 print(matrig_local)
131 print("")
132 print("Matriz de cambio de base Local - Global")
133 print(mat_lg)
134 print("")
135
136 print("Matriz de rigidez en coord. globales")
137 barraA.imprime_mrig()
```

Los resultados obtenidos utilizando el código anterior con los datos mencionados son:

$$\begin{aligned}
 \mathbf{k}_L &= \begin{pmatrix} 656 & | & -656 \\ \hline -656 & | & 656 \end{pmatrix} \\
 \mathbf{M}_{LG} &= \begin{pmatrix} 0.156 & 0 \\ 0.312 & 0 \\ 0.937 & 0 \\ 0 & 0.156 \\ 0 & 0.3126 \\ 0 & 0.937 \end{pmatrix} \\
 \mathbf{k}_L &= \begin{pmatrix} 16 & 32 & 96 & | & -16 & -32 & -96 \\ 32 & 64 & 192 & | & -32 & -64 & -192 \\ 96 & 192 & 576 & | & -96 & -192 & -576 \\ \hline -16 & -32 & -96 & | & 16 & 32 & 96 \\ -32 & -64 & -192 & | & 32 & 64 & 192 \\ -96 & -192 & -576 & | & 96 & 192 & 576 \end{pmatrix}
 \end{aligned}$$

Capítulo 3

El Método Directo de la Rigidez

Índice

3.1. Planteamiento del problema	43
3.2. El Método Directo de la Rigidez	46
3.3. Aplicación de las condiciones de contorno y resolución del sistema	49
3.4. Cálculo de esfuerzos y reacciones	50
3.5. Propiedades de la matriz de rigidez	53
3.6. Pasos para la resolución de problemas	54

3.1. Planteamiento del problema

Como se ha visto en el apartado 1.2, del Capítulo 1, el Cálculo Matricial se basa en aplicar un **proceso de discretización** a la estructura para idealizarla como un conjunto de elementos unidos mediante nudos.

Para definir una estructura, de cara a plantear las ecuaciones del modelo necesitamos determinar lo siguiente:

- El tipo estructural (A2D, P2D, A3D,...).
- La posición de los nudos con respecto a un sistema de coordenadas de referencia.
- La conectividad de los elementos, es decir, nudo inicial y final de cada una de ellas.
- Las propiedades de los materiales y las secciones de los elementos.
- Las condiciones de apoyo.

- Las cargas actuantes.

El objetivo del cálculo estructural es encontrar la respuesta de una estructura dada ante unas acciones definidas. Las acciones vamos a representarlas por un vector \mathbf{F} de $n = N \cdot \text{ngdl}$ componentes, siendo N el número total de nudos, y ngdl el número de gdl s por cada nudo¹, que contendrá las fuerzas aplicadas en cada uno de los nudos. Por ejemplo, para el caso de un P2D en cada nudo hay $\text{ngdl} = 3 \text{ gdl}$ de modo que el *vector de fuerzas* será:

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \vdots \\ \mathbf{F}_N \end{pmatrix} = \begin{pmatrix} F_{1x} \\ F_{1y} \\ M_1 \\ \vdots \\ F_{2x} \\ F_{2y} \\ M_2 \\ \vdots \\ F_{Nx} \\ F_{Ny} \\ M_N \end{pmatrix}$$

La respuesta de la estructura será la deformada que se produzca, es decir los desplazamientos en los nudos, y que podemos recoger en un *vector de desplazamientos* \mathbf{u} del mismo tamaño $N \cdot \text{ngdl}$ de la siguiente forma,

$$\mathbf{u} = \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_N \end{pmatrix} = \begin{pmatrix} u_{1x} \\ u_{1y} \\ \theta_1 \\ \vdots \\ u_{2x} \\ u_{2y} \\ \theta_2 \\ \vdots \\ u_{Nx} \\ u_{Ny} \\ \theta_N \end{pmatrix}$$

¹Puede ocurrir que diferentes nudos tengan un número distinto de ngdl pero no es necesario entrar en esos detalles en este momento de la exposición.

para el caso, igualmente, de un P2D.

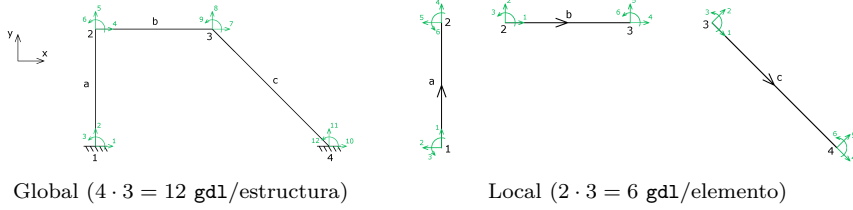


Figura 3.1: Grados de libertad de P2D en los sistemas global y local.

En el diagrama de la figura 3.2 se representan todas las variables del problema en coordenadas globales. Las fuerzas aplicadas en los nudos serán conocidas², mientras que esfuerzos en barras \mathbf{P}^e , desplazamientos en barras δ^e y desplazamientos en nudos \mathbf{u} son desconocidos³. En total habrá $N \cdot \text{ngdl} + 4 \cdot E \cdot \text{ngdl} = (N + 4E) \cdot \text{ngdl}$ incógnitas, siendo E el número de barras de la estructura; ngdl incógnitas en cada nudo y ngdl en ambos extremos de cada barra. Como veremos más adelante, el número total de ecuaciones es el mismo.



Figura 3.2: Diagrama de Tonti para la estructura discretizada.

En el MDR vamos a obtener la relación directa entre *datos* a nivel de estructura, las fuerzas \mathbf{F} , y las *incógnitas* a nivel de estructura, los desplazamientos \mathbf{u} , de la forma,

$$\mathbf{F} = \mathbf{K}\mathbf{u}$$

que será un sistema de $n \times n$ incógnitas, siendo $n = N \cdot \text{ngdl}$, y donde \mathbf{K} se denomina *matriz de rigidez de la estructura*. Para ello, usando las ecuaciones de compatibilidad, comportamiento y equilibrio, eliminaremos las variables a nivel de barra, \mathbf{P}^e y δ^e , para obtener la relación directa entre \mathbf{F} y \mathbf{u} . Las ecuaciones del sistema vamos a establecerlas en el siguiente orden: compatibilidad, comportamiento y finalmente, equilibrio, como se indica en la Figura 3.2.

²Salvo las reacciones, como veremos más adelante en la sección 3.3

³Salvo aquellos que sean conocidos por las condiciones de contorno, como veremos más adelante en la misma sección 3.3

Veamos paso a paso la aplicación de estas ecuaciones, y cómo se obtendrá el sistema de ecuaciones de rigidez global a que da lugar.

3.2. El Método Directo de la Rigidez

Los pasos para obtener el sistema global de rigidez $\mathbf{F} = \mathbf{K}\mathbf{u}$ se describen a continuación.

Ecuaciones de compatibilidad

La compatibilidad de los desplazamientos de la estructura implica que los extremos de las barras que llegan a un nudo k se moverán lo mismo que el propio nudo. Esto es la mera afirmación de que las barras que están unidas antes de aplicar las cargas, lo estarán después de la deformación: es una afirmación meramente geométrica. Las ecuaciones que por tanto han de cumplirse son,

$$\mathbf{u}_k = \boldsymbol{\delta}_k^{e_1} = \boldsymbol{\delta}_k^{e_2} = \boldsymbol{\delta}_k^{e_3} = \dots \quad \forall k = 1 \dots N \quad (3.1)$$

siendo $e_l \ni k$ todos los elementos que confluyen en el nudo k . Otra forma de expresar estas ecuaciones es barra a barra, pues en ambos extremos habrá de cumplirse que,

$$\begin{aligned} \boldsymbol{\delta}_i^e &= \mathbf{u}_i \\ \boldsymbol{\delta}_j^e &= \mathbf{u}_j \end{aligned}$$

para todo $e = 1, \dots, E$, siendo i y j los nudos de la barra e . Visto de esta manera está claro que habrá un total de $2 \cdot E \cdot \text{ngdl}$ ecuaciones.

Ecuaciones constitutivas / ley de comportamiento

En cada barra de la estructura se cumplirá la siguiente relación,

$$\begin{pmatrix} P_i^e \\ P_j^e \end{pmatrix} = \begin{pmatrix} k_{ii}^e & k_{ij}^e \\ k_{ji}^e & k_{jj}^e \end{pmatrix} \begin{pmatrix} \delta_i^e \\ \delta_j^e \end{pmatrix}$$

donde se ha añadido el superíndice “ e ” para indicar el elemento e al que se aplica la relación de rigidez. Esta expresión es aplicable a todas las barras independientemente de su posición, y la obtenemos a partir de las equivalentes en coordenadas locales, mediante las transformaciones que se han visto en el capítulo 2.

Estas ecuaciones, que suman un total de $2 \cdot E \cdot \text{ngdl}$, vamos a escribirlas para cada uno de los nodos, multiplicando por bloques:

$$\begin{aligned} P_i^e &= k_{ii}^e \delta_i^e + k_{ij}^e \delta_j^e \\ P_j^e &= k_{ji}^e \delta_i^e + k_{jj}^e \delta_j^e \end{aligned}$$

pues será necesario para el proceso de reducción al sistema global, como veremos a continuación.

Nótese que la fuerza elemental en un nudo, por ejemplo el \mathbf{P}_i^e tiene dos sumandos: la fuerza debida al movimiento del propio nudo i ($\mathbf{k}_{ii}^e \delta_i^e$), y la debida al movimiento del nudo j ($\mathbf{k}_{ij}^e \delta_j^e$). Los subíndices de las matrices indican precisamente eso: así \mathbf{k}_{ij}^e es la *fuerza* en el nudo i , debido a un *movimiento unidad* en el nudo j .

Ecuaciones de equilibrio

Las ecuaciones de equilibrio se han de cumplir en cada nudo de la estructura:

$$\mathbf{F}_i = \sum_{e \ni i} \mathbf{P}_i^e \quad \forall i = 1 \dots N \quad (3.2)$$

donde la expresión $e \ni i$ indica de nuevo que el sumatorio se extiende a todos elementos e que confluyen en el nudo i . Puesto que en cada nudo hay `ngdl` grados de libertad, el total de ecuaciones de equilibrio es $N \cdot \text{ngdl}$.

Montaje del sistema de ecuaciones global

El conjunto de ecuaciones anteriores suma un total de $(N + 4E) \cdot \text{ngdl}$, que es igual al número de incógnitas. Para obtener el sistema de ecuaciones global $\mathbf{F} = \mathbf{K}\mathbf{u}$, procedemos a eliminar las variables \mathbf{P}^e y δ^e realizando los siguientes pasos:

Eliminación de δ^e : Aplicando las ecuaciones de compatibilidad, se eliminan los desplazamientos δ_i^e y δ_j^e de la relación $\mathbf{P}^e = \mathbf{k}^e \delta^e$ sustituyéndolos por los desplazamientos en los nudos \mathbf{u}_i y \mathbf{u}_j , respectivamente. La ley de comportamiento quedará entonces de la forma,

$$\begin{aligned} \mathbf{P}_i^e &= \mathbf{k}_{ii}^e \mathbf{u}_i + \mathbf{k}_{ij}^e \mathbf{u}_j \\ \mathbf{P}_j^e &= \mathbf{k}_{ji}^e \mathbf{u}_i + \mathbf{k}_{jj}^e \mathbf{u}_j \end{aligned}$$

En este proceso hemos usado las $2 \cdot E \cdot \text{ngdl}$ ecuaciones de compatibilidad para eliminar δ^e incógnitas, que también son $2 \cdot E \cdot \text{ngdl}$.

Eliminación de \mathbf{P}^e : Aplicamos las ecuaciones de equilibrio 3.2, para eliminar las variables \mathbf{P}^e , sustituyendo sus expresiones obtenidas más arriba,

$$\mathbf{F}_i = \sum_{e \ni i} (\mathbf{k}_{ii}^e \mathbf{u}_i + \mathbf{k}_{ij}^e \mathbf{u}_j) \quad \forall i = 1 \dots N \quad (3.3)$$

El resultado es un total de $N \cdot \text{ngdl}$ ecuaciones, para $N \cdot \text{ngdl}$ incógnitas. Estas ecuaciones podemos escribirlas de forma matricial de manera *directa*⁴

⁴Y de aquí proviene el nombre de *directo* de este método matricial

observando que la submatriz elemental \mathbf{k}_{ij}^e aparece la i -ésima ecuación, multiplicando a la j -ésima incógnita, y por tanto la matriz global del sistema vendrá dada por:

$$\mathbf{K} = \begin{pmatrix} \sum_{e \ni 1} \mathbf{k}_{11}^e & \mathbf{k}_{12}^a & \dots & \mathbf{k}_{1i}^b & \dots & \mathbf{k}_{1n}^c \\ \mathbf{k}_{21}^a & \sum_{e \ni 2} \mathbf{k}_{22}^e & \dots & \mathbf{k}_{2i}^d & \dots & \mathbf{k}_{2n}^f \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{k}_{i1}^b & \mathbf{k}_{i2}^d & \dots & \sum_{e \ni i} \mathbf{k}_{ii}^e & \dots & \mathbf{k}_{in}^g \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{k}_{n1}^c & \mathbf{k}_{n2}^f & \dots & \mathbf{k}_{ni}^g & \dots & \sum_{e \ni n} \mathbf{k}_{nn}^e \end{pmatrix} \quad (3.4)$$

Cabe destacar:

- Cada elemento ii de la diagonal principal está formado por la suma de las submatrices \mathbf{k}_{ii}^e de las diferentes barras que confluyen en el nudo i .
- Todo elemento e que une dos nudos $i - j$ tendrá sus 4 submatrices en la matriz de rigidez, y se *ensamblarán o montarán* en las posiciones ii , jj , ij y ji . En la expresión 3.4 podemos comprobar que el elemento a une 1 y 2, el b , 2 e i , el c , 1 y n , el d , 2 e i , el f , 2 y n , y finalmente el g , i y n .
- Como consecuencia de lo anterior, cuando no exista ninguna barra en la estructura que conecte dos nudos i y j , las componentes de la matriz de rigidez \mathbf{K} en las posiciones ij y ji serán nulas.

Podemos ver por tanto, que las submatrices \mathbf{k}_{ij}^e de cada elemento pueden *ensamblarse directamente* sobre la matriz global \mathbf{K} , simplemente conociendo la numeración de los nudos. Por ejemplo, para la estructura P2D de la figura 3.1 la matriz de rigidez global se *ensamblará* a partir de las matrices de rigidez elementales de la forma siguiente,

$$\mathbf{K} = \begin{pmatrix} \mathbf{k}_{11}^a & \mathbf{k}_{12}^a & \mathbf{0} & \mathbf{0} \\ \mathbf{k}_{21}^a & \mathbf{k}_{22}^a + \mathbf{k}_{12}^b & \mathbf{k}_{23}^b & \mathbf{0} \\ \mathbf{0} & \mathbf{k}_{32}^b & \mathbf{k}_{33}^b + \mathbf{k}_{33}^c & \mathbf{k}_{34}^c \\ \mathbf{0} & \mathbf{0} & \mathbf{k}_{43}^c & \mathbf{k}_{44}^c \end{pmatrix} \quad (3.5)$$

donde se han coloreado de forma distinta las submatrices de cada barra, para mayor claridad.

3.3. Aplicación de las condiciones de contorno y resolución del sistema

Aunque más arriba hemos dicho que el vector de desplazamiento \mathbf{u} , que tiene $N \cdot \text{ngdl}$ componentes, englobaba a todas las incógnitas, no todas las componentes serán desconocidas pues los **apoyos** fuerzan que algunas serán conocidas, e igual a cero⁵. A cambio, por cada **gdl** donde el desplazamiento sea conocido, tendremos una fuerza, la reacción, que será una incógnita. En conjunto, por tanto, habrá efectivamente $N \cdot \text{ngdl}$ incógnitas, la mayoría los desplazamientos en los nudos, salvo aquellos que están restringidos, junto con las reacciones precisamente en esos **gdl** restringidos.

Para estructuras de barras planas, los tipos de apoyo más comunes (y sus reacciones correspondientes) se presentan en la Figura 3.3.

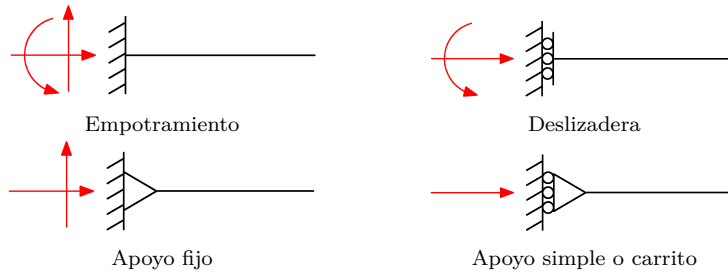


Figura 3.3: Tipos de apoyos comunes en estructuras planas.

En los nudos que no sean apoyos, las únicas fuerzas serán las cargas externas aplicadas en la estructura, que serán conocidas.

Si agrupamos los **gdl**s de la estructura en dos conjuntos, una parte A que incluye a todos los desplazamientos desconocidos (y por tanto las fuerzas son conocidas), y el resto, un conjunto B que contiene a los **gdl**s donde las fuerzas son desconocidas (los apoyos, donde los desplazamientos son conocidos), es posible **particionar** el sistema $\mathbf{F} = \mathbf{K}\mathbf{u}$ de la siguiente forma,

$$\begin{pmatrix} \mathbf{F}_A \\ \mathbf{R}_B \end{pmatrix} = \begin{pmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{BA} & \mathbf{K}_{BB} \end{pmatrix} \begin{pmatrix} \mathbf{u}_A \\ \mathbf{u}_B \end{pmatrix}$$

En estas ecuaciones \mathbf{u}_A y \mathbf{R}_B (las reacciones) son desconocidos, mientras que \mathbf{u}_B y \mathbf{F}_A son conocidos.

Resolvemos el problema seleccionado en primer lugar las ecuaciones del bloque A donde las fuerzas son conocidas,

$$\mathbf{F}_A = \mathbf{K}_{AA}\mathbf{u}_A + \mathbf{K}_{AB}\mathbf{u}_B$$

⁵Salvo en el caso de asientos en apoyos y apoyos elásticos, que veremos cómo tratar en los capítulos 4 y 5.

de donde despejamos \mathbf{u}_A y se obtiene,

$$\mathbf{u}_A = \mathbf{K}_{AA}^{-1}(\mathbf{F}_A - \mathbf{K}_{AB}\mathbf{u}_B) \quad (3.6)$$

que serán los desplazamientos buscados de la estructura. Si los desplazamientos en los apoyos son nulos, como es lo habitual, entonces se reduce a,

$$\mathbf{u}_A = \mathbf{K}_{AA}^{-1}\mathbf{F}_A \quad (3.7)$$

Tras la obtención de los desplazamientos de la estructura (\mathbf{u}_A) se pueden calcular las reacciones \mathbf{R}_B sin tener que resolver un nuevo sistema de ecuaciones mediante,

$$\mathbf{R}_B = \mathbf{K}_{BA}\mathbf{u}_A + \mathbf{K}_{BB}\mathbf{u}_B \quad (3.8)$$

que se reduce a,

$$\mathbf{R}_B = \mathbf{K}_{BA}\mathbf{u}_A \quad (3.9)$$

cuando los desplazamientos en los apoyos son nulos. En la próxima sección veremos una manera alternativa de expresar el cálculo de las reacciones.

3.4. Cálculo de esfuerzos y reacciones

Una vez conocidos los desplazamientos en los nudos, los **esfuerzos** en las barras se pueden calcular aplicando las ecuaciones de comportamiento de cada elemento. Para conocer los esfuerzos es necesario tener el vector \mathbf{P}_L^e , esto es, las fuerzas en los extremos de la barra en coordenadas locales. Para obtenerlas podemos aplicar dos métodos diferentes: o bien se transforman los desplazamientos obtenidos a coordenadas locales

$$\delta_L^e = \mathbf{M}_{GL}\delta^e$$

y a continuación se aplica la ley de comportamiento en locales,

$$\mathbf{P}_L^e = \mathbf{k}_L^e\delta_L^e$$

o bien, se obtienen primeramente las fuerzas en el elemento en globales mediante,

$$\mathbf{P}^e = \mathbf{k}^e\delta^e$$

y se transforman a locales,

$$\mathbf{P}_L^e = \mathbf{M}_{GL}\mathbf{P}^e$$

En los ejercicios resueltos en este texto se opera de este segundo modo.

Las **reacciones** se pueden obtener de dos formas:

1. Aplicando la ecuación 3.8 o la 3.9, si los desplazamientos en los apoyos son nulos.

- Sumando las fuerzas, en globales, de todas las barras que confluyen en el nudo donde vamos a calcular las reacciones:

$$\mathbf{R}_i = \sum_{e \ni i} \mathbf{P}_i^e$$

Este segundo método será el más conveniente, y por ello es necesario calcular los esfuerzos primero en globales, y pasarlos después a locales.

Veamos un ejemplo de aplicación de estos pasos para resolver una estructura por el Método Directo de la Rigidez.

Ejercicio 3.1 Plantear el sistema matricial de ecuaciones $\mathbf{F} = \mathbf{K}\mathbf{u}$ para la siguiente estructura P2D:

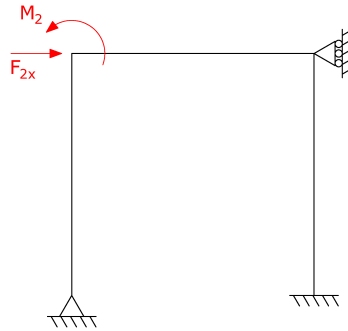


Figura 3.4: Ejercicio 3.1. Pórtico 2D.

Solución

En primer lugar, discretizamos la estructura en elementos y establecemos los sistemas de coordenadas global y locales de cada barra. Además, numeramos los nodos (1-4) y los elementos (a, b, c).

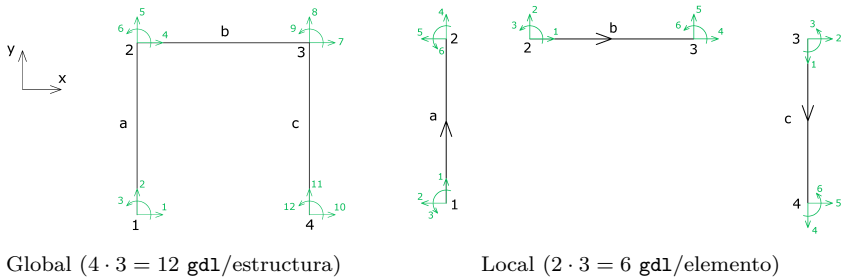


Figura 3.5: Ejercicio 3.1. Discretización y sistemas de coordenadas.

Plantemos las ecuaciones de equilibrio elementales de cada barra, $\mathbf{P}^e = \mathbf{k}^e \delta^e$:

■ Barra a:

$$\begin{pmatrix} P_1^a \\ P_2^a \end{pmatrix} = \begin{pmatrix} k_{11}^a & k_{12}^a \\ k_{21}^a & k_{22}^a \end{pmatrix} \begin{pmatrix} \delta_1^a \\ \delta_2^a \end{pmatrix} \Rightarrow \begin{cases} P_1^a = k_{11}^a \delta_1^a + k_{12}^a \delta_2^a \\ P_2^a = k_{21}^a \delta_1^a + k_{22}^a \delta_2^a \end{cases}$$

■ Barra b:

$$\begin{pmatrix} P_2^b \\ P_3^b \end{pmatrix} = \begin{pmatrix} k_{22}^b & k_{23}^b \\ k_{32}^b & k_{33}^b \end{pmatrix} \begin{pmatrix} \delta_2^b \\ \delta_3^b \end{pmatrix} \Rightarrow \begin{cases} P_2^b = k_{22}^b \delta_2^b + k_{23}^b \delta_3^b \\ P_3^b = k_{32}^b \delta_2^b + k_{33}^b \delta_3^b \end{cases}$$

■ Barra c:

$$\begin{pmatrix} P_3^b \\ P_4^b \end{pmatrix} = \begin{pmatrix} k_{33}^b & k_{34}^b \\ k_{43}^b & k_{44}^b \end{pmatrix} \begin{pmatrix} \delta_3^b \\ \delta_4^b \end{pmatrix} \Rightarrow \begin{cases} P_3^b = k_{33}^b \delta_3^b + k_{34}^b \delta_4^b \\ P_4^b = k_{43}^b \delta_3^b + k_{44}^b \delta_4^b \end{cases}$$

Por compatibilidad de desplazamientos en los nudos:

$$\mathbf{u}_1 = \delta_1^a; \quad \mathbf{u}_2 = \delta_2^a = \delta_2^b; \quad \mathbf{u}_3 = \delta_3^b = \delta_3^c; \quad \mathbf{u}_4 = \delta_4^c$$

Como se ha visto con anterioridad, sustituyendo las ecuaciones de compatibilidad en las de comportamiento y aplicando las ecuaciones de equilibrio, se obtiene la relación $\mathbf{F}_i = \sum_{j=1}^n (\mathbf{k}_{ij} \mathbf{u}_j)$. Para cada nudo de la estructura de este ejemplo:

- Nudo 1: $\mathbf{F}_1 = \mathbf{P}_1^a = \mathbf{k}_{11}^a \mathbf{u}_1 + \mathbf{k}_{12}^a \mathbf{u}_2$
- Nudo 2: $\mathbf{F}_2 = \mathbf{P}_2^a + \mathbf{P}_2^b = \mathbf{k}_{21}^a \mathbf{u}_1 + (\mathbf{k}_{22}^a + \mathbf{k}_{22}^b) \mathbf{u}_2 + \mathbf{k}_{23}^b \mathbf{u}_3$
- Nudo 3: $\mathbf{F}_3 = \mathbf{P}_3^b + \mathbf{P}_3^c = \mathbf{k}_{32}^b \mathbf{u}_2 + (\mathbf{k}_{33}^b + \mathbf{k}_{33}^c) \mathbf{u}_3 + \mathbf{k}_{34}^c \mathbf{u}_4$
- Nudo 4: $\mathbf{F}_4 = \mathbf{P}_4^c = \mathbf{k}_{43}^c \mathbf{u}_3 + \mathbf{k}_{44}^c \mathbf{u}_4$

Expresándolo en forma matricial, se obtiene la ecuación buscada $\mathbf{F} = \mathbf{K} \mathbf{u}$:

$$\begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{pmatrix} = \begin{pmatrix} k_{11}^a & k_{12}^a & 0 & 0 \\ k_{21}^a & k_{22}^{a+b} & k_{23}^b & 0 \\ 0 & k_{32}^b & k_{33}^{b+c} & k_{34}^c \\ 0 & 0 & k_{43}^c & k_{44}^c \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix}$$

donde se emplea la notación simplificada $\mathbf{k}_{ii}^{h+g} = \mathbf{k}_{ii}^h + \mathbf{k}_{ii}^g$.

Si además tenemos en cuenta las condiciones de contorno (fuerzas externas y apoyos), podemos expandir el sistema anterior mostrando los términos de los vectores \mathbf{F}_i y \mathbf{u}_i :

$$\begin{pmatrix} R_{1x} \\ R_{1y} \\ 0 \\ F_{2x} \\ 0 \\ M_2 \\ R_{3x} \\ 0 \\ 0 \\ R_{4x} \\ R_{4y} \\ M_4 \end{pmatrix} = \begin{pmatrix} k_{11}^a & k_{12}^a & 0 & 0 \\ k_{21}^a & k_{22}^{a+b} & k_{23}^b & 0 \\ 0 & k_{32}^b & k_{33}^{b+c} & k_{34}^c \\ 0 & 0 & k_{43}^c & k_{44}^c \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \theta_1 \\ u_{2x} \\ u_{2y} \\ \theta_2 \\ 0 \\ u_{3y} \\ \theta_3 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Lógicamente, las matrices k_{ij} y 0 tienen dimensiones 3×3 , al ser en este caso $ngdl = 3$, el número de *gdl* de los nudos.

Dado que las fuerzas externas F_{2x} y M_2 son conocidas (deberán ser datos del problema), se puede observar que este sistema tiene un total de 12 ecuaciones y 12 incógnitas (6 reacciones y 6 desplazamientos), por lo que está determinado. Para resolverlo se podrá particionar seleccionando las ecuaciones 3, 4, 5, 6, 8 y 9, que son aquellas que tienen como incógnita los desplazamientos.

En la práctica, una vez conocido el funcionamiento del método, para la resolución de problemas es posible plantear directamente este sistema matricial observando la estructura, y comenzar así con él la resolución del problema.

3.5. Propiedades de la matriz de rigidez

La matriz de rigidez cumple las siguientes propiedades:

- **Simétrica:** $K_{ij} = K_{ji}$. Las fuerzas producidas en el *gdl* i al desplazar una unidad el *gdl* j es la misma que se produce en el *gdl* j al desplazar una unidad el *gdl* i (Teorema de Reciprocidad).
- **En banda.** Al ser nulos los elementos de la matriz correspondientes a *gdl* no conectados, con una correcta numeración de los nudos se consigue concentrar todos los elementos no nulos sobre una banda junto a la diagonal principal. Esta propiedad permite que, en la resolución computacional de estructuras a través del Método de la Rigidez, se ensamble y almacene únicamente la banda central de la matriz, ahorrando así memoria y tiempo de cálculo.

- **Definida positiva.** El menor de los autovalores de la matriz, tras aplicar las condiciones de contorno, es mayor que cero. Físicamente, esto indica que es un conjunto indeformable, sin mecanismos internos y con condiciones de apoyo suficientes para impedir el desplazamiento como sólido rígido.
- **Diagonalmente dominante.** Todos los términos de la diagonal principal son positivos e iguales o mayores a la suma del resto de elementos de su fila o columna. Esta condición no se cumple estrictamente.

Todas estas propiedades hacen que el sistema de ecuaciones resultante $\mathbf{F} = \mathbf{K}\mathbf{u}$ pueda resolverse de forma computacionalmente muy eficaz, lo que implica que para unos recursos informáticos dados (memoria y CPU), pueda encontrarse la solución para un número muy alto de gdl.

3.6. Pasos para la resolución de problemas

La resolución manual de problemas mediante el MDR tiene como inconveniente el implicar la resolución de operaciones y sistemas matriciales de cierta extensión. Sin embargo, al tratarse de un método orientado al uso computacional, el procedimiento de resolución de los ejercicios resulta muy mecánico, pudiendo ser aplicado de manera casi inalterada a cualquier tipo de estructura.

Veamos de forma resumida los pasos principales para la resolución de problemas mediante el MDR.

1. Analizar los **datos del problema** y definir las características de la estructura:
 - Tipo de estructura (A2D, P2D, A3D,...).
 - Coordenadas de los nudos con respecto a un sistema de referencia global y numeración de nudos. Una adecuada numeración de los nudos ayudará a tener una matriz de rigidez más ordenada con la que resolver fácilmente el sistema.
 - Elementos que conectan los nudos (barras y tirantes). Nombrarlos de la forma más ordenada posible e indicar un sentido para sus ejes locales.
 - Definir las propiedades de los materiales y de las secciones de los elementos. Para las estructuras P2D harán falta el módulo de elasticidad (E), el área (A) y la inercia (I) de la sección. Es útil calcular los productos EA y EI , pues siempre aparecen estos productos en las matrices. Recuerde que hay que respetar la coherencia en las unidades empleadas.
 - Condiciones de apoyo y fuerzas externas actuantes. Permitirán determinar qué términos son conocidos y cuáles son incógnitas en

los vectores de fuerzas y desplazamientos, así como las fuerzas de empotramiento a calcular (ver Capítulo 4).

2. Definir los **sistemas de coordenadas** global, y local de cada elemento. La posición de los nudos y la orientación del elemento, es decir, la elección de un nudo como inicial y otro como final, determina los ejes locales⁶. Es importante señalar que la matriz final del sistema \mathbf{K} no depende de la orientación ni numeración de las barras. Con la elección de los sistemas de coordenadas se tendrán definidos los **gd1** de la estructura y de cada elemento.
3. **Montar la matriz de rigidez global \mathbf{K}** . El tamaño de esta matriz será $n \times n$, siendo $n = N \cdot \text{ngd1}$ el número total de **gd1** de la estructura.
4. **Particionar** el sistema seleccionando las filas A donde las fuerzas sean conocidas (\mathbf{F}_A) y las columnas A y B correspondientes a los desplazamientos desconocidos (\mathbf{u}_A) y conocidos (\mathbf{u}_B) para obtener el sistema reducido 3.6, cuya resolución nos dará los desplazamientos incógnita del problema \mathbf{u}_A .
5. **Montar el vector de fuerzas \mathbf{F}** . Rellenar el vector \mathbf{F} con las fuerzas conocidas en cada nudo. Además, como veremos en el Capítulo 4, cuando haya acciones que no están aplicadas directamente en los nudos (fuerzas repartidas en las barras, temperaturas, etc), habrá que calcular las **fuerzas de empotramiento** en las barras y montarlas en el vector $\tilde{\mathbf{F}}$.
6. Para cada barra, calcular las **submatrices \mathbf{k}_{ij}^e** que intervienen en el sistema a resolver. Se calcularán directamente en coordenadas globales, siempre que tengamos la expresión de la misma.
7. Resolver el sistema particionado 3.6 para obtener los **desplazamientos**. En caso de que hubiera desplazamientos condensados (ver Capítulo 5), estos se calcularían con posterioridad.
8. Calcular los **esfuerzos** en las barras a partir de las ecuaciones de comportamiento ($\mathbf{P}^e = \mathbf{k}^e \delta^e$; $\mathbf{P}_L^e = \mathbf{M}_{GL} \mathbf{P}^e$).
9. Calcular las **reacciones** en los apoyos como suma de los esfuerzos \mathbf{P}_i^e que llegan a cada apoyo i ($\mathbf{R}_i = \sum_{e \in i} \mathbf{P}_i^e$).
10. **Comprobar los resultados:** la estructura como conjunto, cada barra y cada nudo deben estar en equilibrio.
11. **Dibujar la deformada:** También sirve como comprobación, para ver si es razonable.

⁶Esto es así todos los casos, salvo los P3D donde hará falta un punto adicional para fijar los ejes locales y y z

Ejercicio 3.2 Calcular los desplazamientos en los nudos, los esfuerzos en las barras y las reacciones en los apoyos de la siguiente estructura mediante el Método Directo de la Rigidez.

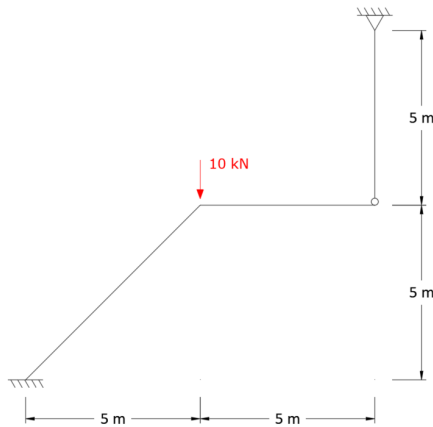


Figura 3.6: Ejercicio 3.2. Estructura planteada.

Datos: $E = 2 \cdot 10^6 \text{ kN/m}^2$; $A = 0.2 \text{ m}^2$; $I = 1.8 \cdot 10^{-3} \text{ m}^4$.

Solución

Comenzamos definiendo el tipo de estructura que se nos presenta en el problema. Es posible observar que las barras inclinada y horizontal se unen entre sí rígidamente, mientras que la barra vertical se une con la horizontal mediante una rótula. Además, esta barra tiene liberado también el giro en el apoyo superior, por lo que trabajará sólo a axil, es decir, se trata de una barra articulada bidimensional (tirante). De este modo, podemos determinar que las barras inclinada y horizontal son de tipo P2D mientras que la barra vertical es de tipo A2D.

Identificado el tipo de cada barra, numeramos los nodos (1-4) y los elementos (a, b, c) y establecemos los sistemas de coordenadas global y locales de cada barra, identificando los respectivos *gdl*s.

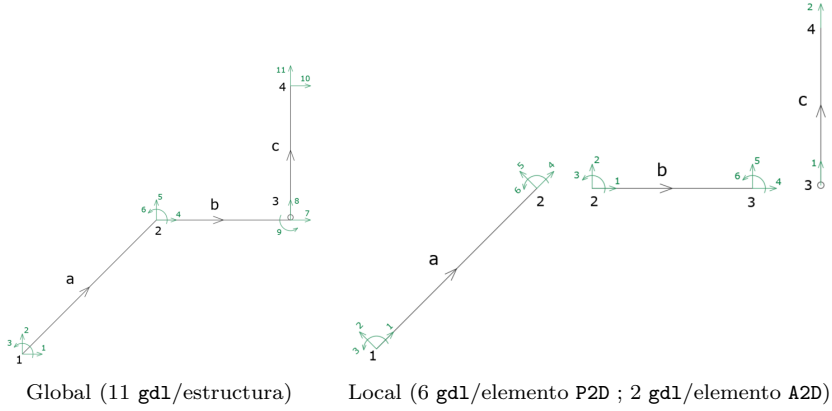


Figura 3.7: Ejercicio 3.2. Discretización y sistema de coordenadas.

Observamos que el **gdl 9** en coordenadas globales corresponde al giro en el nudo 3 sólo de la barra b, ya que el tirante c tiene los giros liberados. La barra c, al ser de tipo **A2D** tiene 2 gdl/nudo en coordenadas globales y sólo 1 gdl/nudo en coordenadas locales.

Planteamos el sistema de ecuaciones de la estructura, de la forma $\mathbf{F} = \mathbf{K}\mathbf{u}$:

$$\begin{pmatrix} R_{1x} \\ R_{1y} \\ M_1 \\ 0 \\ -10 \\ 0 \\ 0 \\ 0 \\ 0 \\ R_{4x} \\ R_{4y} \end{pmatrix} = \begin{pmatrix} k_{11}^a & k_{12}^a & 0 & 0 \\ k_{21}^a & k_{22}^{a+b} & k_{23}^b & 0 \\ 0 & k_{32}^b & k_{33}^{b+c} & k_{34}^c \\ 0 & 0 & k_{43}^c & k_{44}^c \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ u_{2x} \\ u_{2y} \\ \theta_2 \\ u_{3x} \\ u_{3y} \\ \theta_3^b \\ 0 \\ 0 \end{pmatrix}$$

Cabe destacar que todas las matrices de rigidez son de dimensiones 3×3 excepto $\mathbf{k}_{34}^c (3 \times 2)$, $\mathbf{k}_{43}^c (2 \times 3)$ y $\mathbf{k}_{44}^c (2 \times 2)$, al ser la barra c de tipo **A2D**.

Para calcular los desplazamientos debemos particionar el sistema seleccionando las ecuaciones que tienen a los desplazamientos como incógnitas (ecuaciones de la cuarta a la novena). De forma compacta podemos expresar:

sarlo como:

$$\begin{pmatrix} \mathbf{F}_2 \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{k}_{22}^{a+b} & \mathbf{k}_{23}^b \\ \mathbf{k}_{32}^b & \mathbf{k}_{33}^{b+c} \end{pmatrix} \begin{pmatrix} \mathbf{u}_2 \\ \mathbf{u}_3 \end{pmatrix}$$

Calculamos en primer lugar los elementos necesarios de la matriz de rigidez.

- Barra a (P2D; $L = 5\sqrt{2}$ m; $\alpha = 45^\circ$)

$$\mathbf{k}^a = \begin{pmatrix} \mathbf{k}_{11}^a & \mathbf{k}_{12}^a \\ \mathbf{k}_{21}^a & \mathbf{k}_{22}^a \end{pmatrix} \Rightarrow \mathbf{k}_{22}^a = \begin{pmatrix} 28345.37 & 28223.18 & 305.47 \\ 28223.18 & 28345.37 & -305.47 \\ 305.47 & -305.47 & 2036.47 \end{pmatrix}$$

- Barra b (P2D; $L = 5$ m; $\alpha = 0^\circ$)

$$\mathbf{k}^b = \begin{pmatrix} \mathbf{k}_{22}^b & \mathbf{k}_{23}^b \\ \mathbf{k}_{32}^b & \mathbf{k}_{33}^b \end{pmatrix} =$$

$$= \begin{pmatrix} 8 \cdot 10^4 & 0 & 0 & -8 \cdot 10^4 & 0 & 0 \\ 0 & 345.6 & 864 & 0 & -345.6 & 864 \\ 0 & 864 & 2880 & 0 & -864 & 1140 \\ -8 \cdot 10^4 & 0 & 0 & 8 \cdot 10^4 & 0 & 0 \\ 0 & -345.6 & -864 & 0 & 345.6 & -864 \\ 0 & 864 & 1140 & 0 & -864 & 2880 \end{pmatrix}$$

- Barra c (A2D; $L = 5$ m; $\alpha = 90^\circ$)

$$\mathbf{k}^c = \begin{pmatrix} \mathbf{k}_{33}^c & \mathbf{k}_{34}^c \\ \mathbf{k}_{43}^c & \mathbf{k}_{44}^c \end{pmatrix} \Rightarrow \mathbf{k}_{33}^c = \begin{pmatrix} 0 & 0 \\ 0 & 8 \cdot 10^4 \end{pmatrix}$$

Sumamos los términos correspondientes. Nótese que para sumar las matrices \mathbf{k}_{33}^b y \mathbf{k}_{33}^c es necesario añadir una fila y columna de ceros en el *gdI* del giro de la barra c.

$$\mathbf{k}_{22}^{a+b} = \mathbf{k}_{22}^a + \mathbf{k}_{22}^b = \begin{pmatrix} 108345.37 & 28223.18 & 305.47 \\ 28223.18 & 28690.97 & 558.53 \\ 305.47 & 558.53 & 4916.47 \end{pmatrix}$$

$$\mathbf{k}_{33}^{b+c} = \begin{pmatrix} 8 \cdot 10^4 & 0 & 0 \\ 0 & 345.6 & -864 \\ 0 & -864 & 2880 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 8 \cdot 10^4 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 8 \cdot 10^4 & 0 & 0 \\ 0 & 80345.6 & -864 \\ 0 & -864 & 2880 \end{pmatrix}$$

Podemos ahora montar el sistema y calcular los desplazamientos:

$$\begin{pmatrix} 0 \\ -10 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 108345.37 & 28223.18 & 305.47 \\ 28223.18 & 28690.97 & 558.53 \\ 305.47 & 558.53 & 4916.47 \\ -8 \cdot 10^4 & 0 & 0 \\ 0 & -345.6 & -864 \\ 0 & 864 & 1140 \end{pmatrix} \begin{pmatrix} -8 \cdot 10^4 & 0 & 0 \\ 0 & -345.6 & 864 \\ 0 & -864 & 1140 \\ 8 \cdot 10^4 & 0 & 0 \\ 0 & 80345.6 & -864 \\ 0 & -864 & 2880 \end{pmatrix} \begin{pmatrix} u_{2x} \\ u_{2y} \\ \theta_2 \\ u_{3x} \\ u_{3y} \\ \theta_3^b \end{pmatrix}$$

Resolviendo se obtiene:

$$\begin{aligned} u_{2x} &= 3.088 \cdot 10^{-2} \text{ m}; & u_{2y} &= -3.100 \cdot 10^{-2} \text{ m}; & \theta_2 &= -1.317 \cdot 10^{-3} \text{ rad} \\ u_{3x} &= 3.090 \cdot 10^{-2} \text{ m}; & u_{3y} &= -4.055 \cdot 10^{-5} \text{ m}; & \theta_3^b &= 9.946 \cdot 10^{-3} \text{ rad} \end{aligned}$$

Pasamos ahora a calcular los esfuerzos en las barras.

■ Barra a.

Calculamos la matriz \mathbf{k}^a para la barra a:

$$\begin{pmatrix} \mathbf{P}_1^a \\ \mathbf{P}_2^a \end{pmatrix} = \begin{pmatrix} \mathbf{k}_{11}^a & \mathbf{k}_{12}^a \\ \mathbf{k}_{21}^a & \mathbf{k}_{22}^a \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 = 0 \\ \mathbf{u}_2 \end{pmatrix} \Rightarrow \mathbf{P}_2^a = \mathbf{k}_{22}^a \mathbf{u}_2 = \begin{pmatrix} 0 \\ -6.76 \\ 16.22 \end{pmatrix}$$

Se ha resuelto la segunda línea del sistema, ya que para ella bastaba utilizar la matriz \mathbf{k}_{22}^a que ya teníamos previamente calculada. Las fuerzas en el nudo 1 se pueden obtener a través de la expresión anterior como $\mathbf{P}_1^a = \mathbf{k}_{12}^a \mathbf{u}_2$ o bien por equilibrio en la barra una vez calculado \mathbf{P}_2^a ; el resultado en cualquier caso es el que se muestra en la siguiente figura.

Los resultados están en coordenadas globales, para calcular los esfuerzos deben pasarse a coordenadas locales de la barra mediante la matriz de cambio de coordenadas global-local ($\mathbf{P}_{i,L}^a = \mathbf{m}_{GL} \mathbf{P}_{i,L}^a$) o por trigonometría.

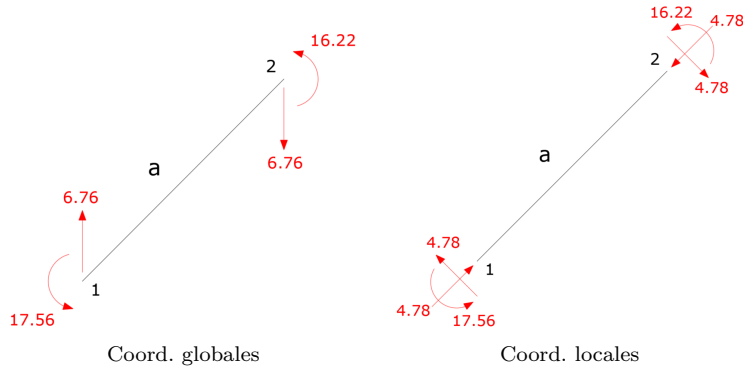


Figura 3.8: Ejercicio 3.2. Fuerzas en los nudos de la barra a [kN, kN · m].

Se obtienen las siguientes leyes de esfuerzos:

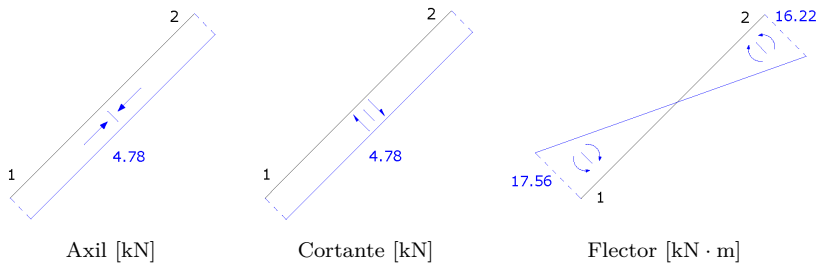


Figura 3.9: Ejercicio 3.2. Diagramas de esfuerzos en la barra a .

■ Barra b .

Seguimos el mismo procedimiento que para la barra a .

$$\begin{pmatrix} P_2^b \\ P_3^b \end{pmatrix} = \begin{pmatrix} k_{22}^b & k_{23}^b \\ k_{32}^b & k_{33}^b \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \end{pmatrix} \Rightarrow P_2^b = k_{22}^b u_2 + k_{23}^b u_3 = \begin{pmatrix} 0 \\ -3.24 \\ -16.22 \end{pmatrix}$$

Gráficamente:

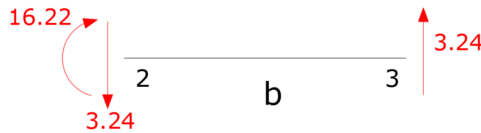


Figura 3.10: Ejercicio 3.2. Fuerzas en los nudos de la barra b [kN, kN · m].

Las leyes de esfuerzos que se muestran a continuación, siendo nulo, como debe ser, el flector en la rótula (nudo 3).

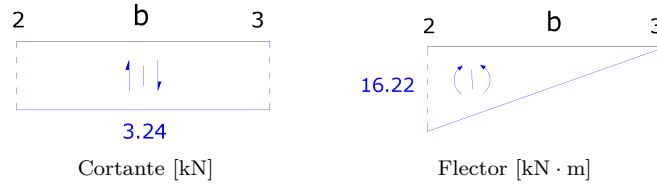


Figura 3.11: Ejercicio 3.2. Diagramas de esfuerzos en la barra b .

■ Barra c .

$$\begin{pmatrix} P_3^c \\ P_4^c \end{pmatrix} = \begin{pmatrix} k_{33}^c & k_{34}^c \\ k_{43}^c & k_{44}^c \end{pmatrix} \begin{pmatrix} u_3 \\ u_4 = 0 \end{pmatrix} \Rightarrow P_3^c = k_{33}^c u_3 = \begin{pmatrix} 0 \\ -3.24 \end{pmatrix}$$

Hay que tener en cuenta que, como estamos trabajando con una barra $A2D$, la matriz de rigidez k_{33}^c es (2×2) y el vector u_3 tiene sólo componentes de desplazamiento vertical y horizontal, no de giro.

Representamos gráficamente estas fuerzas nodales y los esfuerzos correspondientes:

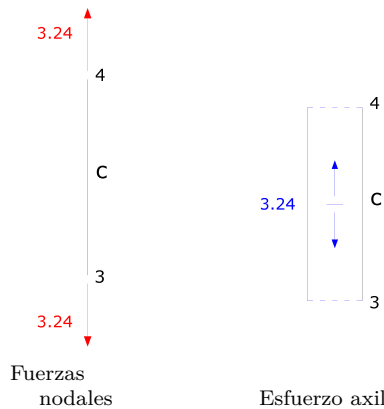


Figura 3.12: Ejercicio 3.2. Fuerzas nodales y diagrama de esfuerzos en la barra c [kN].

El tirante c trabaja solo a axil, a tracción en este caso.

Por último, calculamos las **reacciones** en los apoyos, que serán iguales a la suma de las fuerzas nodales de las barras que llegan a dichos apoyos.

En este caso:

$$\mathbf{F}_1 = \mathbf{P}_1^a = \begin{pmatrix} 0 \\ 6.76 \\ 17.56 \end{pmatrix}; \quad \mathbf{F}_4 = \mathbf{P}_4^c = \begin{pmatrix} 0 \\ 3.24 \\ 0 \end{pmatrix}$$

Lo representamos gráficamente y comprobamos que la estructura está en equilibrio.

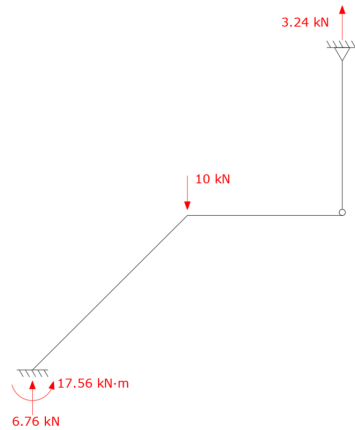


Figura 3.13: Ejercicio 3.2. Reacciones y cargas en la estructura.

Ejercicio Python 3.1

Se considera la siguiente estructura compuesta por dos barras:

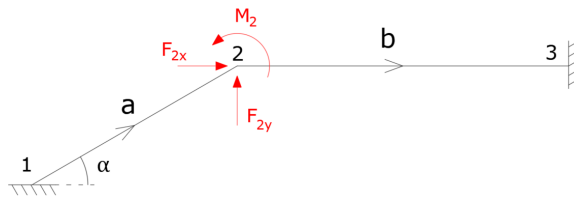


Figura 3.14: Ejercicio Python 3.1. Estructura P2D.

Se pide calcular los desplazamientos en el nudo 2 mediante el Método Directo de la Rigidez, teniendo en cuenta los siguientes datos:

- Propiedades del material: $E = 20 \text{ GPa}$.
- Barra a : longitud 5 m, sección cuadrada de lado 10 cm, formando un ángulo con la horizontal $\alpha = 30^\circ$.

- Barra b : longitud 7 m, sección cuadrada de lado 20 cm, horizontal.
- Fuerzas actuantes: $F_{2x} = 100 \text{ kN}$, $F_{2y} = -200 \text{ kN}$, $M_2 = 300 \text{ kN} \cdot \text{m}$.

El código que se muestra a continuación define en primer lugar dos funciones: una primera función “K_P2D” para calcular la matriz de rigidez 6×6 de un pórtico 2D en coordenadas globales, y otra función “K_P2D_sub” que usa la anterior para obtener la submatriz 3×3 (\mathbf{k}_{ii} , \mathbf{k}_{ij} , \mathbf{k}_{ji} ó \mathbf{k}_{jj}) de un pórtico 2D.

Definidas las funciones, se introducen los datos de las barras y se usa la función “K_P2D_sub” para obtener las submatrices necesarias. Se ensamblan las submatrices, se introduce el vector de fuerzas y se resuelve el sistema $\mathbf{F} = \mathbf{K}\mathbf{u}$ con los desplazamientos como incógnitas.

```

1
2 from __future__ import division
3 import numpy as np
4
5 #=====
6 # FUNCIONES PARA P2D
7 #=====
8 #-----
9 # Función K_P2D: Matriz de rigidez pórtico bidimensional P2D
10 #-----
11 def K_P2D(EA, EI, L, Angulo):
12     ## Función que obtiene la matriz de rigidez 6x6, en coordenadas
13     ## globales, del P2D
14     ## EA = Producto EA de la barra
15     ## EI = Producto EI de la barra
16     ## L = Longitud de la barra
17     ## Angulo = angulo de orientación de la barra, en grados
18     ## sexagesimales
19     Angulo = Angulo * np.pi / 180.
20
21     # Matriz de cambio de base de local a global M_{L,G}:
22     M_LG = np.array([[np.cos(Angulo), -np.sin(Angulo), 0],
23                      [np.sin(Angulo), np.cos(Angulo), 0],
24                      [0, 0, 1]])
25
26     # Se expande a una 6x6. Se construye por bloques, el bloque
27     # fuera de la diagonal es una matriz 3x3 de ceros. Esto se
28     # genera con np.zeros((3,3))
29     M_LG = np.vstack([np.hstack([M_LG, np.zeros((3,3))]),
30                      np.hstack([np.zeros((3,3)), M_LG])])
31
32     # Primero se inicia en cero:
33     K_L = np.zeros((6,6))
34
35     # Se rellenan los términos de la diagonal superior:
36     K_L[0,:] = np.array([EA / L, 0., 0., -EA / L, 0., 0.])
37     K_L[1,1:] = np.array([12 * EI / L**3, 6 * EI / L**2, 0,
38                          -12 * EI / L**3, 6 * EI / L**2])
39     K_L[2,2:] = np.array([4 * EI / L, 0, -6 * EI / L**2, 2 * EI
40                          / L])
41     K_L[3,3:] = np.array([EA / L, 0, 0])

```

3. El Método Directo de la Rigidez

```
37     K_L[4,4:] = np.array([12 * EI / L**3, -6 * EI / L**2])
38     K_L[5,5]  = 4 * EI / L
39
40 # Se construye el resto de la matriz por simetría. Se suma
41 # K_local + traspuesta (K_local), en numpy la traspuesta se
42 # obtiene con .T
43 # Se resta la diagonal de k_local para que no aparezca 2 veces
44 K_L = (K_L + K_L.T) - np.diag ( np.diag(K_L) )
45
46 # Cambio de base a coordenadas globales: K_G = M(L,G) * K_L * M
47 # (G,L)
48 # El producto se realiza sabiendo que el producto matricial A*B
49 # = np.dot(A,B)
50 K_G = np.dot(M_LG, np.dot(K_L , M_LG.T ) )
51
52 # Salida de la función: la submatriz cambiada a globales
53 return K_G
54
55 #-----
56 # Función para definir una submatriz de un P2D
57 #-----
58 def K_P2D_sub(ea, ei, lon, angle, icol, jcol):
59     ## Esta función obtiene la submatriz 3x3 de k_lineal de un P2D
60     ## cambiada a globales
61     ## ea = Producto EA de la barra
62     ## ei = Producto EI de la barra
63     ## lon = Longitud de la barra
64     ## angle = ángulo de orientación de la barra, en grados
65     ## sexagesimales
66     ## icol = posición (puede ser 1 o 2)
67     ## jcol = posición (puede ser 1 o 2)
68     if (icol not in ([1,2]) or jcol not in ([1,2]) ):
69         print("Valores de icol, jcol fuera de rango")
70     return
71
72 # Puesto que Python indexa los vectores y matrices comenzando
73 # con el cero, indicaremos las posiciones restando 1 a las
74 # posiciones que se numeran comenzando desde el 1
75 icol = icol -1
76 jcol = jcol -1
77
78 # Llamada a la función K_P2D para generar la matriz 6x6:
79 K_globales = K_P2D(ea, ei, lon, angle)
80
81 # Extracción del bloque:
82 K_Bloque = K_globales[icol * 3 : icol * 3 + 3 , jcol * 3 :
83 jcol * 3 + 3]
84
85 return K_Bloque
86
87 #=====
88 # RESOLUCIÓN DEL PROBLEMA
89 # Uso de funciones para definir submatrices en P2D
90 #=====
```

```

84
85 #-----
86 # Definición de las barras
87 #-----
88 Elast = 20E6 # 20 GPa = 20E+6 MPa, común para las dos barras
89
90 # Barra a
91 Aa = 0.1**2 # Sección cuadrada de 0.1 m de lado
92 Ia = 1./12 * 0.1**4 # Inercia en m^4
93 La = 5 # Longitud en metros
94 alfa_a = 30 # Ángulo de la barra: 30 grados
95
96 EA_a = Elast * Aa
97 EI_a = Elast * Ia
98
99
100 # Barra b
101 Ab = 0.2**2 # Sección cuadrada, de 0.2 m de lado
102 Ib = 1./12 * 0.2**4 # Inercia en m^4
103 Lb = 7 # Longitud en metros
104 alfa_b = 0 # Ángulo de la barra b
105
106 EA_b = Elast * Ab
107 EI_b = Elast * Ib
108
109 #-----
110 # Submatrices para el nudo 2
111 #-----
112 # Para la barra a, el nudo 2 es el nudo final, por lo que se
113 # debe extraer la submatriz del bloque "2,2":
114 k22_a = K_P2D_sub(EA_a, EI_a, La, alfa_a, 2, 2)
115
116 # Para la barra b, el nudo 2 es el nudo inicial, por lo que se
117 # debe extraer la submatriz del bloque "1,1":
118 k22_b = K_P2D_sub(EA_b, EI_b, Lb, alfa_b, 1, 1)
119
120 # Mostramos estas submatrices
121 print("K22_a =")
122 print(k22_a)
123 print("")
124 print("K22_b =")
125 print(k22_b)
126
127 #-----
128 # Ensamblaje: K22_a + K22_b
129 #-----
130 k22 = k22_a + k22_b
131 print("")
132 print("K22_a + k22_b =")
133 print(k22)
134
135 #-----
136 # Vector de fuerzas en el nodo 2
137 #-----
138 fuerzas = [100, -200, 300]

```

3. El Método Directo de la Rigidez

```
138 # -----
139 # Resolución del sistema de ecuaciones
140 # Obtención de desplazamientos
141 # -----
142 u_2 = np.linalg.solve(k22, fuerzas)
143 print("")
144 print("Solución del sistema ")
145 print("u_2x =", u_2[0], "m")
146 print("u_2y =", u_2[1], "m")
147 print("theta_2 =", u_2[2], "rad")
```

El resultado obtenido ejecutando el código es el siguiente:

$$u_{2x} = 4.644 \cdot 10^{-3} \text{ m}; \quad u_{2y} = -3.314 \cdot 10^{-2} \text{ m}; \quad \theta_2 = 1.868 \cdot 10^{-1} \text{ rad}$$

Como ejercicio adicional, se propone modificar el código anterior para resolver la siguiente estructura, donde las barras tienen módulo de elasticidad $E = 20 \text{ GPa}$ y sección cuadrada de lado 10 cm en el caso de la barra a y de lado 25 cm para la barra b .

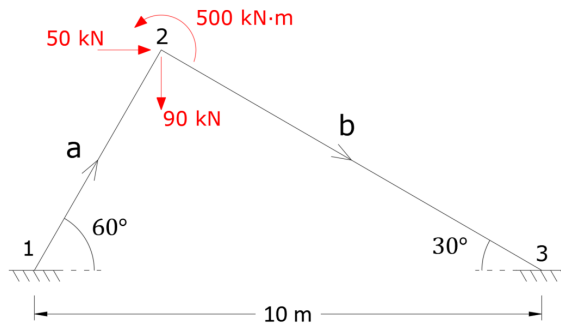


Figura 3.15: Ejercicio Python 3.1. Estructura P2D - Variante propuesta.

Resolviendo correctamente el ejercicio se obtienen los siguientes desplazamientos:

$$u_{2x} = -1.205 \cdot 10^{-3} \text{ m}; \quad u_{2y} = -3.20 \cdot 10^{-3} \text{ m}; \quad \theta_2 = 1.159 \cdot 10^{-1} \text{ rad}$$

Capítulo 4

Cargas aplicadas en barras y otras acciones

Índice

4.1. Introducción	67
4.2. Cargas aplicadas en barras	68
4.3. Cargas térmicas	72
4.4. Asientos en apoyos	73

4.1. Introducción

Para aplicar el Método Directo de la Rigidez tal y como se ha explicado en los capítulos anteriores, se ha considerado la existencia únicamente de cargas exteriores aplicadas en los nudos. Sin embargo, es evidente que en la realidad existen cargas que se aplican directamente sobre los elementos de la estructura (peso propio, sobrecargas...), por lo que será necesario disponer de un procedimiento para incorporar este tipo de acciones de forma sistemática al procedimiento del Cálculo Matricial.

El MDR, como hemos visto, trabaja con vectores de fuerzas y desplazamientos en los nudos, por lo que la forma de tratar las acciones sobre las barras será “transformándolas” en cargas en los extremos de las mismas y que sean estáticamente equivalentes a las existentes. Esto se hace mediante la introducción del concepto de **fuerzas de empotramiento**. Llamamos fuerzas de empotramiento producidas por una carga cualquiera q a las reacciones que existirían en una viga biempotrada sujeta a dicha carga q .

En el presente capítulo se van a estudiar tres tipos de acciones, comunes en las estructuras, que no se aplican directamente en los nudos:

- Cargas aplicadas en barras (puntuales o distribuidas).
- Cargas térmicas (uniformes o con un gradiente en la sección).
- Asientos en apoyos (que ocurren en los nudos pero generan esfuerzos en las barras).

4.2. Cargas aplicadas en barras

En el caso de una carga puntual aplicada en un punto intermedio de una barra, cabría la posibilidad de plantearse definir un nudo intermedio en dicho punto y resolver la estructura de la forma vista en el Capítulo 3, pero esto presenta dos inconvenientes: aumenta innecesariamente el número de gdl, y con ello la complejidad del problema, y no es generalizable al caso de cargas distribuidas.

Por este motivo, el método general consiste en descomponer la estructura en dos estados (I y II) y, aplicando el principio de superposición, obtener los resultados del problema como suma de los resultados de los dos estados.

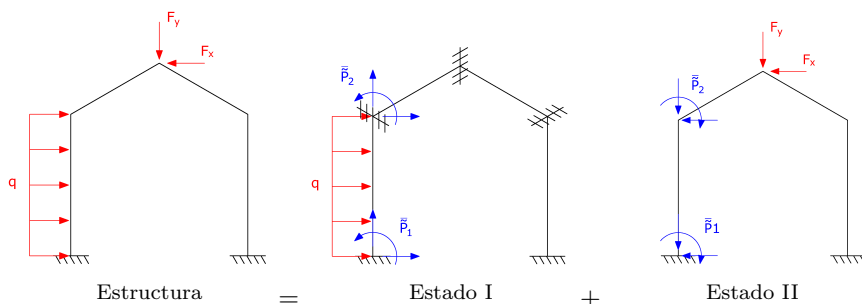


Figura 4.1: Descomposición de la estructura en estados I y II.

Como se puede apreciar en la Figura 4.1, los estados en los que se divide la estructura son:

- **Estado I:** todas barras se consideran biempotradas (no hay desplazamientos en los nudos) y se consideran sólo las acciones en las barras. En los nudos biempotrados aparecerán unas reacciones que llamaremos fuerzas de empotramiento (\vec{P}), que pueden calcularse por los métodos convencionales de Resistencia de Materiales.
- **Estado II:** estructura con condiciones de apoyo originales y fuerzas aplicadas únicamente en los nudos.

Para que se cumpla el **principio de superposición**, debe verificarse que:

- La suma de los desplazamientos de los estados I y II sea igual a los desplazamientos de la estructura original. Dado que los desplazamientos en I son nulos, los desplazamientos resultantes se corresponden con los del estado II.

$$\mathbf{u}^{\text{total}} = \mathbf{u}^I + \mathbf{u}^{II} \Rightarrow \mathbf{u}^{\text{total}} = \mathbf{u}^{II}$$

- La suma de los esfuerzos en los estados I y II sea igual a los esfuerzos en la estructura original.

$$\mathbf{P}^{\text{total}} = \mathbf{P}^I + \mathbf{P}^{II}$$

Para que la suma de fuerzas de I y II sea igual a la original, las fuerzas de empotramiento que aparecen en los nudos del estado I se deberán incluir en el estado II con signo contrario.

Este estado II tiene sólo cargas en los nudos, por lo que se puede resolver por el MDR tal y como se ha explicado en los capítulos anteriores. Si llamamos $\tilde{\mathbf{F}}$ al vector de fuerzas de empotramiento en los nudos, tendremos:

$$\mathbf{F} - \tilde{\mathbf{F}} = \mathbf{K}\mathbf{u}$$

Caso particular: cargas distribuidas inclinadas

Como se ha indicado, las fuerzas de empotramiento se pueden calcular resolviendo las barras biempotradas mediante las ecuaciones de Resistencia de Materiales. Los casos más comunes, de hecho, se pueden encontrar resueltos en multitud de prontuarios de estructuras.

Vamos ahora a ver el caso particular de una barra biempotrada sometida a una carga distribuida no ortogonal a la misma, para intentar asimilarla a los casos recogidos en los formularios y así facilitar la resolución de los problemas.

Se tenga la siguiente barra de longitud L , inclinada un ángulo α con respecto al sistema de coordenadas global, y sometida a una carga uniformemente distribuida q en la dirección negativa del eje y global:

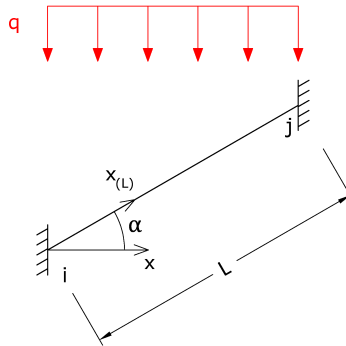


Figura 4.2: Viga biempotrada con carga uniforme inclinada.

4. Cargas aplicadas en barras y otras acciones

Teniendo en cuenta que:

$$x = x_L \cos \alpha \Rightarrow dx = dx_L \cos \alpha$$

Podemos expresar la resultante Q de la carga distribuida q sobre la barra de dos formas distintas:

$$\left. \begin{aligned} Q &= \int_0^{L \cos \alpha} q \, dx = qL \cos \alpha \\ Q &= \int_0^L q_L \, dx_L = q_L L \end{aligned} \right\} \Rightarrow qL \cos \alpha = q_L L \Rightarrow q_L = q \cos \alpha$$

Es decir, la carga vertical q aplicada a lo largo del eje x equivale a una carga vertical de valor $q \cos \alpha$ aplicada a lo largo de la barra. Si lo representamos gráficamente y descomponemos la carga en una acción perpendicular y otra paralela a la barra:

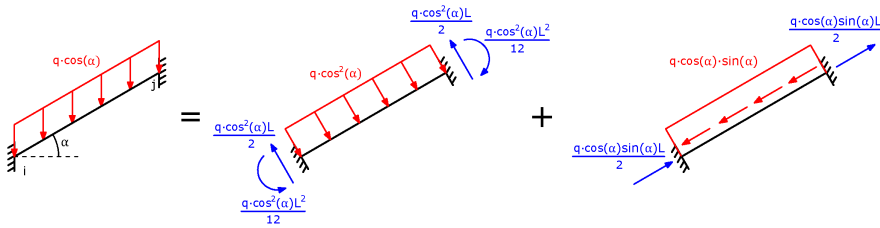


Figura 4.3: Descomposición de carga uniforme inclinada.

En esta figura ya se han representado las reacciones en los empotramientos (fuerzas de empotramiento) que, al haber transformado la carga a casos comunes, se pueden encontrar en cualquier prontuario.

En el sistema de coordenadas global, tenemos:

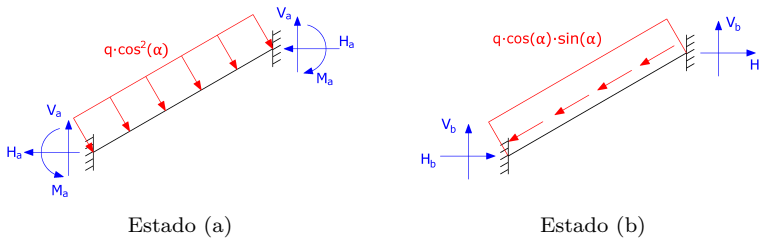


Figura 4.4: Fuerzas de empotramiento en coordenadas globales.

Donde:

$$\begin{aligned} \text{(a): } H_a &= \frac{qL \cos^2 \alpha \sin \alpha}{2}; & V_a &= \frac{qL \cos^3 \alpha}{2}; & M_a &= \frac{qL^2 \cos^2 \alpha}{12} \\ \text{(b): } H_b &= \frac{qL \sin \alpha \cos^2 \alpha}{2}; & V_b &= \frac{qL \sin^2 \alpha \cos \alpha}{2} \end{aligned}$$

Por tanto:

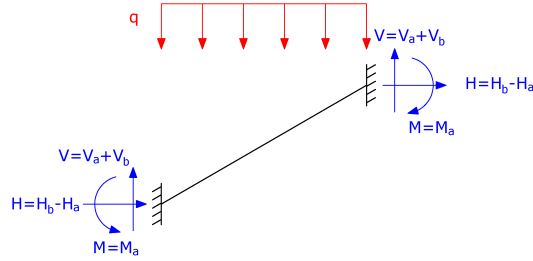


Figura 4.5: Fuerzas de empotramiento resultantes

Que operando, se obtiene:

$$H = 0$$

$$V = \frac{qL \cos \alpha}{2} (\cos^2 \alpha + \sin^2 \alpha) = \frac{qL \cos \alpha}{2}$$

$$M = \frac{qL^2 \cos^2 \alpha}{12}$$

Se puede observar que las resultantes obtenidas corresponden a las de una barra biempotrada de longitud $L \cos \alpha$ sometida a una carga uniformemente distribuida de valor q perpendicular a la su eje:

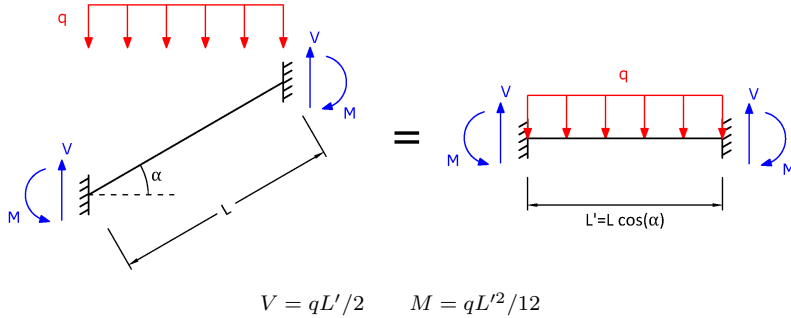


Figura 4.6: Equivalencia con barra de longitud $L \cos \alpha$.

Esto quiere decir que, cuando tengamos una barra biempotrada de longitud L sometida a una carga inclinada de valor q , se podrán obtener las fuerzas de empotramiento como aquellas producidas en una barra biempotrada de longitud $L \cos \alpha$ sometida a una carga ortogonal a su eje.

Esta equivalencia se puede aplicar a cualquier barra biempotrada sometida a cualquier carga inclinada con respecto a su eje, tanto puntual como distribuida.

La equivalencia no será aplicable a barras con otras condiciones de apoyo distintas (como sucede cuando se condensan **gd1**, según se explica en el Capítulo 5). En esos casos, habrá que descomponer la fuerza en una perpendicular y otra paralela a la barra (como en la Figura 4.4), obtener las reacciones de estos dos estados y sumarlas.

4.3. Cargas térmicas

La aparición de gradientes térmicos en las barras de una estructura isostática no genera esfuerzos ni tensiones adicionales en las mismas (siempre que la variación de la temperatura en el canto del elemento sea lineal). Esto no sucede así en el caso de estructuras hiperestáticas, donde en general aparecerán esfuerzos en las barras cuando existan gradientes térmicos.

La forma de tratar estos esfuerzos es análoga a la planteada para las cargas en barras, es decir, se plantea un estado I con empotramientos en todos los nudos y se calculan las reacciones que se producen en el elemento sometido a la carga térmica. Estas reacciones, llamadas fuerzas de empotramiento, se incluyen en el estado II con signo contrario.

Llamaremos T_e y T_i a las temperaturas en la cara exterior e interior de un elemento, respectivamente, T_0 a la temperatura en el centro de gravedad de la sección, h al canto de la barra y α al coeficiente de dilatación del material:

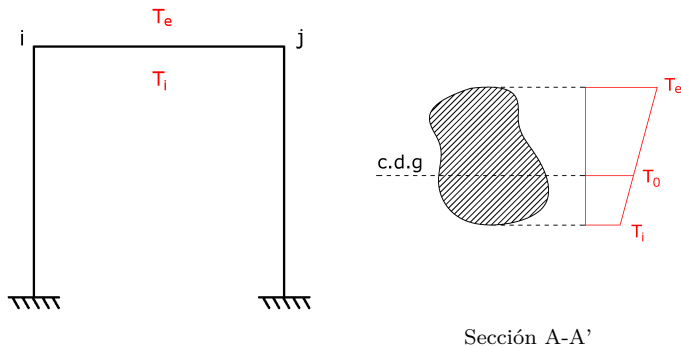


Figura 4.7: Barra sometida a gradiente térmico lineal en su sección.

Consideramos la barra como biempotrada para obtener las fuerzas de empotramiento:

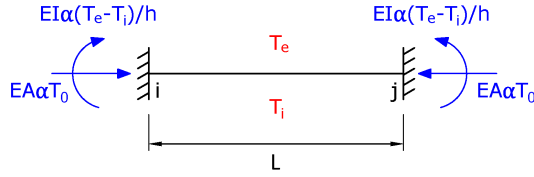


Figura 4.8: Fuerzas de empotramiento en barra sometida a gradiente térmico lineal en su sección.

Es decir, las fuerzas de empotramiento en los nudos i y j debido a la carga térmica serán:

$$\tilde{\mathbf{P}}_i = \begin{pmatrix} EA\alpha T_0 \\ 0 \\ -EI\alpha \frac{T_e - T_i}{h} \end{pmatrix}; \quad \tilde{\mathbf{P}}_j = \begin{pmatrix} -EA\alpha T_0 \\ 0 \\ EI\alpha \frac{T_e - T_i}{h} \end{pmatrix}$$

En el caso en que el aumento de temperatura sea constante en toda la sección ($T_e = T_i = T_0$), en la barra se inducirán solamente esfuerzos axiales (de compresión para incrementos de temperatura):

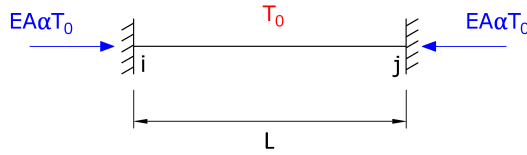


Figura 4.9: Fuerzas de empotramiento en barra sometida a carga térmica constante en su sección.

4.4. Asientos en apoyos

En una estructura hiperestática, el movimiento de un apoyo provoca la aparición de esfuerzos en sus elementos. Para una mayor comodidad en la resolución de problemas, estos esfuerzos serán tratados de la misma forma que las cargas en barras o los gradientes de temperatura, o sea, se incluirán en el sistema como fuerzas de empotramiento.

Veamos cómo obtener estas fuerzas de empotramiento debidas al asiento de un apoyo mediante un ejemplo. Imaginemos que tenemos la siguiente estructura, que sufre un asiento en el apoyo del nudo 4, tal y como se muestra en la figura:

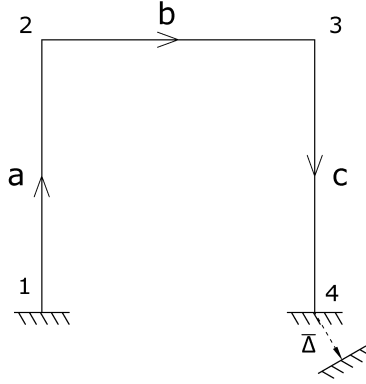


Figura 4.10: Estructura con asiento en apoyo.

El asiento se puede expresar mediante un vector de desplazamientos conocidos Δ , que de forma general tendrá componentes horizontal, vertical y de giro:

$$\Delta = \begin{pmatrix} \Delta_x \\ \Delta_y \\ \Delta_\theta \end{pmatrix}$$

El desplazamiento Δ puede incluirse en el vector u del sistema de ecuaciones,

$$\begin{pmatrix} R_1 \\ 0 \\ 0 \\ R_4 \end{pmatrix} = \begin{pmatrix} k_{11}^a & k_{12}^a & 0 & 0 \\ k_{21}^a & k_{22}^{a+b} & k_{23}^b & 0 \\ 0 & k_{32}^b & k_{33}^{b+c} & k_{34}^c \\ 0 & 0 & k_{43}^c & k_{44}^c \end{pmatrix} \begin{pmatrix} 0 \\ u_2 \\ u_3 \\ \Delta \end{pmatrix}$$

Como queremos que el desplazamiento Δ aparezca como una fuerza de empotramiento, operamos hasta obtener:

$$\begin{pmatrix} R_1 \\ 0 \\ 0 \\ R_4 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ k_{34}^c \\ k_{44}^c \end{pmatrix} \Delta = \begin{pmatrix} k_{11}^a & k_{12}^a & 0 & 0 \\ k_{21}^a & k_{22}^{a+b} & k_{23}^b & 0 \\ 0 & k_{32}^b & k_{33}^{b+c} & k_{34}^c \\ 0 & 0 & k_{43}^c & k_{44}^c \end{pmatrix} \begin{pmatrix} 0 \\ u_2 \\ u_3 \\ 0 \end{pmatrix}$$

Por tanto, el asiento en el nudo 4 generará fuerzas de empotramiento en la barra c (única barra que conecta con dicho nudo), de valor:

$$\begin{aligned} \tilde{P}_3^c &= k_{34}^c \Delta \\ \tilde{P}_4^c &= k_{44}^c \Delta \end{aligned}$$

A continuación se muestra un ejemplo de resolución de estructura con los tres tipos de acciones en barras vistos en el presente capítulo.

Ejercicio 4.1 *El pórtico de la siguiente figura está sometido a una carga de viento dominante en su cara oeste de 2 kN/m, un aumento de la temperatura homogéneo en su cubierta de 30 °C debido a la acción del sol, y una mala cimentación provoca un asiento en su apoyo derecho de 20 cm hacia abajo.*

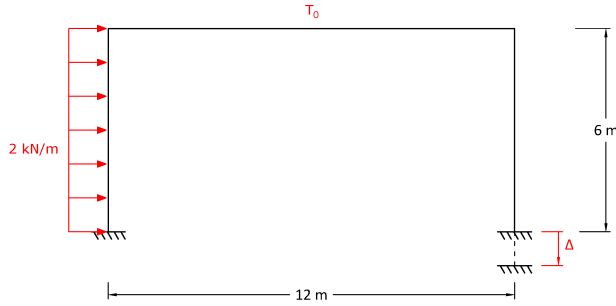


Figura 4.11: Ejercicio 4.1. Pórtico con acciones en barras.

Se pide calcular el desplazamiento en los nudos y los esfuerzos en la barra horizontal.

Datos: $EA = 2 \cdot 10^6$ kN; $EI = 3 \cdot 10^4$ kNm²; $\alpha = 10^{-5}$ °C⁻¹.

Solución

Establecemos los sistemas de coordenadas global y locales, definiendo los gdl en cada caso:

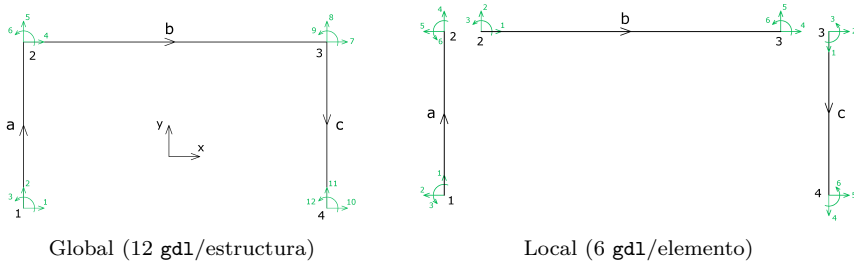


Figura 4.12: Ejercicio 4.1. Sistemas de coordenadas y gdl.

Planteamos el sistema matricial de ecuaciones $\mathbf{F} - \tilde{\mathbf{F}} = \mathbf{K}\mathbf{u}$ teniendo en cuenta que tenemos las siguientes fuerzas de empotramiento:

- Barra a (nudos 1 y 2): fuerza de empotramiento debida a carga distribuida.

- Barra b (nudos 2 y 3): fuerza de empotramiento debida a carga térmica.
- Barra c (nudos 3 y 4): fuerza de empotramiento debida a asiento en apoyo 4.

El sistema queda:

$$\begin{pmatrix} R_{1x} \\ R_{1y} \\ M_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ R_{4x} \\ R_{4y} \\ M_4 \end{pmatrix} - \begin{pmatrix} \tilde{P}_1^a \\ \vdots \\ \tilde{P}_2^{a+b} \\ \vdots \\ \tilde{P}_3^{b+c} \\ \vdots \\ \tilde{P}_4^c \end{pmatrix} = \begin{pmatrix} k_{11}^a & k_{12}^a & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ k_{21}^a & k_{22}^{a+b} & k_{23}^b & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & k_{32}^b & k_{33}^{b+c} & k_{34}^c \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & k_{43}^c & k_{44}^c \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ u_{2x} \\ u_{2y} \\ \theta_2 \\ u_{3x} \\ u_{3y} \\ \theta_3 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Para calcular los desplazamientos, particionamos el sistema y seleccionamos las ecuaciones que tienen a estos como incógnitas (ecuaciones de la cuarta a la novena):

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} \tilde{P}_2^{a+b} \\ \tilde{P}_3^{b+c} \end{pmatrix} = \begin{pmatrix} k_{22}^{a+b} & k_{23}^b \\ k_{32}^b & k_{33}^{b+c} \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \end{pmatrix}$$

Calculamos en primer lugar las fuerzas de empotramiento, según se ha explicado en el presente capítulo. Calcularemos sólo las necesarias para resolver el ejercicio, que serán las de los nudos 2 y 3.

- Barra a ($L = 6 \text{ m}$; $q = 2 \text{ kN/m}$)

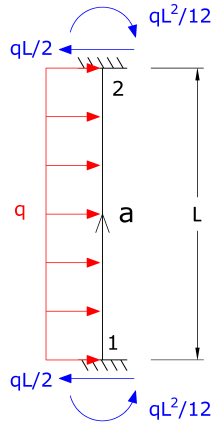


Figura 4.13: Ejercicio 4.1. Fuerzas de empotramiento en barra a.

$$\tilde{\mathbf{P}}_2^a = \begin{pmatrix} -\frac{qL}{2} \\ 0 \\ -\frac{qL^2}{12} \end{pmatrix} = \begin{pmatrix} -6 \\ 0 \\ -6 \end{pmatrix}$$

- Barra b ($L = 12$ m; $T_0 = 30$ °C)

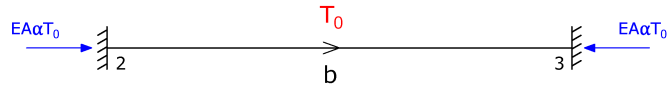


Figura 4.14: Ejercicio 4.1. Fuerzas de empotramiento en barra b.

$$\tilde{\mathbf{P}}_2^b = \begin{pmatrix} EA\alpha T_0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 600 \\ 0 \\ 0 \end{pmatrix}$$

$$\tilde{\mathbf{P}}_3^b = \begin{pmatrix} -EA\alpha T_0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -600 \\ 0 \\ 0 \end{pmatrix}$$

- Barra c ($L = 6 \text{ m}$; $\alpha = 270^\circ$; $\Delta_{4y} = 0.2 \text{ cm}$)

$$\begin{pmatrix} \tilde{P}_3^c \\ \tilde{P}_4^c \end{pmatrix} = \begin{pmatrix} k_{33}^c & k_{34}^c \\ k_{43}^c & k_{44}^c \end{pmatrix} \begin{pmatrix} 0 \\ \Delta \end{pmatrix} \Rightarrow \tilde{P}_3^c = k_{34}^c \Delta = k_{34}^c \begin{pmatrix} 0 \\ -0.2 \\ 0 \end{pmatrix}$$

Calculamos únicamente los términos de k_{34}^c que necesitamos y obtenemos la fuerza de empotramiento:

$$k_{34}^c = \begin{pmatrix} \times & 0 & \times \\ \times & -3.33 \cdot 10^5 & \times \\ \times & 0 & \times \end{pmatrix} \Rightarrow \tilde{P}_3^c = k_{34}^c \Delta = \begin{pmatrix} 0 \\ 6.67 \cdot 10^4 \\ 0 \end{pmatrix}$$

El siguiente paso es obtener los términos de K que intervienen en el sistema:

- Barra a ($L = 6 \text{ m}$; $\alpha = 90^\circ$)

$$k^a = \begin{pmatrix} k_{11}^a & k_{12}^a \\ k_{21}^a & k_{22}^a \end{pmatrix} \Rightarrow k_{22}^a = \begin{pmatrix} 1666.67 & 0 & 5000 \\ 0 & 3.33 \cdot 10^5 & 0 \\ 5000 & 0 & 2 \cdot 10^4 \end{pmatrix}$$

- Barra b ($L = 12 \text{ m}$; $\alpha = 0^\circ$)

$$k^b = \begin{pmatrix} k_{22}^b & k_{23}^b \\ k_{32}^b & k_{33}^b \end{pmatrix} = \begin{pmatrix} 1.67 \cdot 10^5 & 0 & 0 & -1.67 \cdot 10^5 & 0 & 0 \\ 0 & 2.08 \cdot 10^5 & 1250 & 0 & -2.08 \cdot 10^5 & 1250 \\ 0 & 1250 & 10^4 & 0 & -1250 & 5000 \\ -1.67 \cdot 10^5 & 0 & 0 & 1.67 \cdot 10^5 & 0 & 0 \\ 0 & -2.08 \cdot 10^5 & -1250 & 0 & 2.08 \cdot 10^5 & -1250 \\ 0 & 1250 & 5000 & 0 & -1250 & 10^4 \end{pmatrix}$$

- Barra c ($L = 6 \text{ m}$; $\alpha = 270^\circ$)

$$k^c = \begin{pmatrix} k_{33}^c & k_{34}^c \\ k_{43}^c & k_{44}^c \end{pmatrix} \Rightarrow k_{33}^c = \begin{pmatrix} 1666.67 & 0 & 5000 \\ 0 & 3.33 \cdot 10^5 & 0 \\ 5000 & 0 & 2 \cdot 10^4 \end{pmatrix}$$

Finalmente montamos el sistema y calculamos los desplazamientos:

$$-\begin{pmatrix} 594 \\ 0 \\ -6 \\ -600 \\ 6.67 \cdot 10^4 \\ 0 \end{pmatrix} = \begin{pmatrix} 1.68 \cdot 10^5 & 0 & 5000 \\ 0 & 3.34 \cdot 10^5 & 1250 \\ 5000 & 1250 & 3 \cdot 10^4 \\ -1.67 \cdot 10^5 & 0 & 0 \\ 0 & -2.08 \cdot 10^5 & -1250 \\ 0 & 1250 & 5000 \end{pmatrix} \begin{pmatrix} -1.67 \cdot 10^5 & 0 & 0 \\ 0 & -2.08 \cdot 10^5 & 1250 \\ 0 & -1250 & 5000 \\ 1.68 \cdot 10^5 & 0 & 5000 \\ 0 & 3.34 \cdot 10^5 & -1250 \\ 5000 & -1250 & 3 \cdot 10^4 \end{pmatrix} \begin{pmatrix} u_{2x} \\ u_{2y} \\ \theta_2 \\ u_{3x} \\ u_{3y} \\ \theta_3 \end{pmatrix}$$

Se obtiene:

$$\mathbf{u}_2 = \begin{pmatrix} u_{2x} \\ u_{2y} \\ \theta_2 \end{pmatrix} = \begin{pmatrix} 3.84 \cdot 10^{-2} \text{ m} \\ -2.90 \cdot 10^{-5} \text{ m} \\ -1.23 \cdot 10^{-2} \text{ rad} \end{pmatrix}$$

$$\mathbf{u}_3 = \begin{pmatrix} u_{3x} \\ u_{3y} \\ \theta_3 \end{pmatrix} = \begin{pmatrix} 4.20 \cdot 10^{-2} \text{ m} \\ -0.20 \text{ m} \\ -1.33 \cdot 10^{-2} \text{ rad} \end{pmatrix}$$

Hallados los desplazamientos, se determinan los esfuerzos en la barra horizontal (barra b):

$$\begin{pmatrix} \mathbf{P}_2^b \\ \mathbf{P}_3^b \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{P}}_2^b \\ \tilde{\mathbf{P}}_3^b \end{pmatrix} + \begin{pmatrix} \mathbf{k}_{22}^b & \mathbf{k}_{23}^b \\ \mathbf{k}_{32}^b & \mathbf{k}_{33}^b \end{pmatrix} \begin{pmatrix} \mathbf{u}_2 \\ \mathbf{u}_3 \end{pmatrix} \Rightarrow \begin{cases} \mathbf{P}_2^b = \tilde{\mathbf{P}}_2^b + \mathbf{k}_{22}^b \mathbf{u}_2 + \mathbf{k}_{23}^b \mathbf{u}_3 \\ \mathbf{P}_3^b = \tilde{\mathbf{P}}_3^b + \mathbf{k}_{32}^b \mathbf{u}_2 + \mathbf{k}_{33}^b \mathbf{u}_3 \end{cases} \Rightarrow$$

$$\Rightarrow \mathbf{P}_2^b = \begin{pmatrix} 3.59 \\ 9.66 \\ 60.37 \end{pmatrix}; \quad \mathbf{P}_3^b = \begin{pmatrix} -3.59 \\ -9.66 \\ 55.59 \end{pmatrix}$$

Gráficamente:

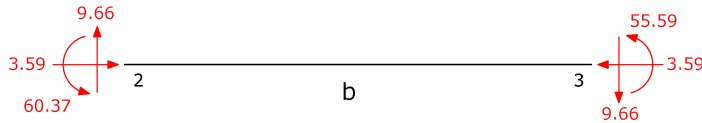


Figura 4.15: Ejercicio 4.1. Fuerzas en los nudos de la barra b [kN, kNm].

Por lo que los diagramas de esfuerzos serán:

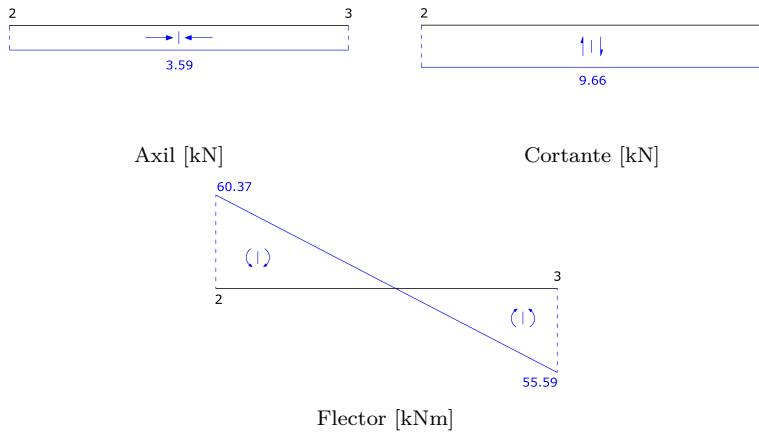


Figura 4.16: Ejercicio 4.1. Diagramas de esfuerzos en la barra b.

Ejercicio Python 4.1

Se considera la siguiente estructura, sometida a una carga distribuida, donde las barras *a* y *b* se unen entre sí rígidamente mientras la barra *c* lo hace a través de una rótula.

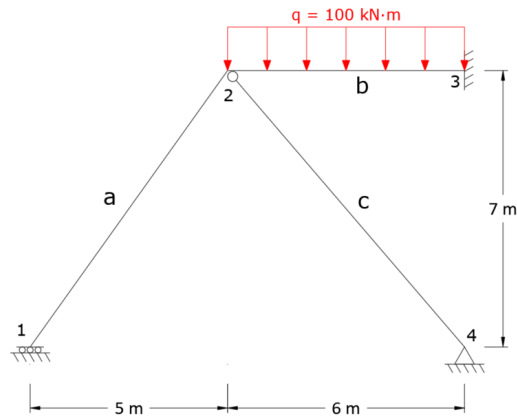


Figura 4.17: Ejercicio Python 4.1. Estructura con carga distribuida.

Se pide calcular los desplazamientos en los nudos mediante el Método Directo de la Rigidez, teniendo en cuenta los siguientes datos:

- Barra *a*: $E_a = 20 \text{ GPa}$, sección cuadrada de lado 10 cm.
- Barra *b*: $E_b = 25 \text{ GPa}$, sección cuadrada de lado 20 cm.

- Barra c : $E_c = 200$ GPa, sección circular de diámetro 5 cm.

El código mostrado a continuación para resolver este problema comienza definiendo una clase “Punto2D” y una función “longitud” para calcular la distancia entre dos puntos, igual que en el ejercicio de [Python 2.1](#). Además se define una función “alfa” que calcula el ángulo de la recta P1-P2 con la horizontal.

Por otra parte se define una clase “P2D” y una clase “A2D”, para barras de pórtico plano y de articulada plana, respectivamente; cada una con las siguientes funciones:

- “K_local” para obtener la matriz de rigidez en coordenadas locales.
- “M_lg” para definir la matriz de cambio de locales a globales.
- “K_global” para obtener la matriz de rigidez en coordenadas globales.
- “K_gobal_caja(i,j)” para obtener la submatriz k_{ij} en globales.

Definidas las funciones, la resolución del problema sigue el siguiente esquema:

1. Definir los nudos de la estructura con la clase “Punto2D”.
2. Definir las barras introduciendo los materiales y usando las clases “P2D” o “A2D”, según corresponda.
3. Calcular las submatrices necesarias y ensamblar la matriz del sistema teniendo en cuenta las condiciones de contorno, es decir, se trabajará con los `gdl` donde los desplazamientos están permitidos.
4. Definir el vector de fuerzas de empotramiento producidas por la carga distribuida q . En este caso no hay cargas nodales aplicadas.
5. Resolver el sistema de ecuaciones para calcular los desplazamientos.

```

1 from __future__ import division
2 import numpy as np
3
4 #=====
5 # DEFINICIÓN DE CLASES Y FUNCIONES
6 # Matrices de Rigidez P2D y A2D en globales
7 # Completa y con submatrices
8 # Programación Orientada a Objetos
9 #=====
10
11 #-----
12 # Definición de la clase "Punto". Coordenadas (x, y) de un punto
13 # Observación: Se inicializa con las coordenadas (0,0)
14 # Se define bidimensional, pues se usará para el P2D
15 #-----
16 class Punto2D:
17     """ Clase para representar los puntos en 3D, coordenadas x,
        y, z """

```

4. Cargas aplicadas en barras y otras acciones

```
18     def __init__(self, x=0, y=0):
19         self.x = x
20         self.y = y
21     def coords(self):
22         return "{0}, {1}".format(self.x, self.y)
23
24 #-----
25 # Definición de la función "longitud"
26 # Calcula la distancia entre dos puntos
27 # Trabaja con p1,p2. Es una función "amiga" de la clase Punto
28 #-----
29 def longitud(p1, p2):
30     """ Devuelve la longitud (módulo) del vector p1-p2 """
31     Longitud = 0
32     Longitud += (p2.x - p1.x ) ** 2
33     Longitud += (p2.y - p1.y ) ** 2
34     Longitud = np.sqrt(Longitud)
35     return Longitud
36
37 #-----
38 # Definición de la función "alfa"
39 # Devuelve el ángulo que define el vector P1 -> P2
40 # La función "arctan2" devuelve el ángulo buscado
41 #-----
42 def alfa(p1,p2):
43     """Angulo P1 -> P2"""
44     vy = p2.y -p1.y
45     vx = p2.x -p1.x
46     angulo = np.arctan2(vy,vx)
47     return angulo
48
49 #-----
50 # Definición de la clase "P2D", para barra P2D
51 #-----
52 # Inicialización: A partir de dos puntos, y los parámetros EA,
53 #   EI
54 # Funciones de la clase:
55 #   K_local: Para obtener la matriz de rigidez en locales (6x6
56 #       )
57 #   M_lg: Matriz de cambio de base de Locales a Globales (6x6)
58 #   K_global: Para obtener la matriz de rigidez en globales (6
59 #       x6)
60 #   K_global_caja(i,j): Para obtener la submatriz ij en
61 #       globales (3x3)
62
63 class P2D:
64     """ Clase para representar la barra P2D """
65     def __init__(self, p1 = Punto2D(), p2 = Punto2D(), ea = 0,
66         ei = 0):
67         self.p1 = p1
68         self.p2 = p2
69         self.ea = ea
70         self.ei = ei
71
72     def K_local(self):
73         L = longitud(self.p1,self.p2)
```

```

69     EA = self.ea
70     EI = self.ei
71
72     K_L = np.zeros((6,6))
73
74     K_L[0,:] = np.array([ EA / L, 0., 0., - EA / L, 0., 0.])
75     K_L[1,1:] = np.array([12 * EI / L**3, 6* EI / L**2, 0,
76                           -12 * EI / L**3, 6 * EI / L**2])
77     K_L[2,2:] = np.array([4 * EI / L, 0, -6 * EI / L**2, 2 *
78                             EI / L ])
79     K_L[3,3:] = np.array([EA / L, 0, 0])
80     K_L[4,4:] = np.array([12 * EI / L**3, -6 * EI / L**2])
81     K_L[5,5] = 4 * EI / L
82
83     K_L = (K_L + K_L.T) - np.diag ( np.diag(K_L) )
84
85     return K_L
86
87 def M_lg(self):
88     Angulo = alfa (self.p1, self.p2)
89
90     M_LG = np.array([[np.cos(Angulo), -np.sin(Angulo), 0],
91                     [np.sin(Angulo), np.cos(Angulo), 0],
92                     [0, 0, 1]])
93     M_LG = np.vstack([np.hstack([M_LG, np.zeros((3,3))]),
94                       np.hstack([np.zeros((3,3)), M_LG])])
95
96     return M_LG
97
98 def K_global(self):
99
100     M_LG = self.M_lg()
101     K_L = self.K_local()
102
103     K_G = np.dot(M_LG, np.dot(K_L , M_LG.T ))
104
105     return K_G
106
107 # Función para extraer una submatriz
108 def K_global_caja(self,icol=1,jcol=1):
109     if (icol not in ([1,2]) or jcol not in ([1,2])):
110         print("Valores de icol, jcol fuera de rango")
111         return
112     icol = icol -1
113     jcol = jcol -1
114     kg = self.K_global()
115
116     K_sub = kg[icol * 3:icol * 3 + 3 , jcol * 3 : jcol * 3 +
117               3]
118
119     return K_sub
120
121 # Definición de la clase "A2D", para barra A2D
122 # Inicialización: A partir de dos puntos, y el parámetro ea

```

4. Cargas aplicadas en barras y otras acciones

```
123 # Funciones de la clase:
124 #     K_local : Para obtener la matriz de rigidez en locales (2
125 #         x2)
126 #     M_lg : Matriz de cambio de base de Locales a Globales (4x2
127 #         )
128 #     K_global: Para obtener la matriz de rigidez en globales (6
129 #         x6)
130 #     K_global_caja(i,j): Para obtener la submatriz en ij en
131 #         globales (2x2)
132
133 class A2D:
134     """ Clase para representar la barra A2D """
135     def __init__(self, p1 = Punto2D(), p2 = Punto2D(), ea = 0):
136         self.p1 = p1
137         self.p2 = p2
138         self.ea = ea
139
140     def K_local(self):
141         L = longitud(self.p1, self.p2)
142         EA = self.ea
143         K_L = np.array([[EA / L, -EA / L],
144                         [-EA / L, EA / L]])
145         return K_L
146
147     def M_lg(self):
148         Angulo = alfa (self.p1, self.p2)
149         M_LG = np.array([[np.cos(Angulo), 0],
150                         [np.sin(Angulo), 0],
151                         [0, np.cos(Angulo)],
152                         [0, np.sin(Angulo)]])
153         return M_LG
154
155     def K_global(self):
156         M_LG = self.M_lg()
157         K_L = self.K_local()
158         K_G = np.dot(M_LG, np.dot(K_L, M_LG.T ))
159         return K_G
160
161 # Función para extraer una submatriz
162 def K_global_caja(self, icol=1, jcol=1):
163     if (icol not in ([1,2]) or jcol not in ([1,2]) ):
164         print("Valores de icol, jcol fuera de rango")
165         return
166     icol = icol -1
167     jcol = jcol -1
168     kg = self.K_global()
169     K_sub = kg[icol * 2:icol * 2 + 2 , jcol * 2 : jcol * 2 +
170 2]
171     return K_sub
172
173 #=====
174 # RESOLUCIÓN DEL PROBLEMA
175 # Cálculo de estructura P2D - A2D
176 #=====
```



```

174 #-----
175 # Definición de los nudos
176 #-----
177 p1 = Punto2D(0,0)
178 p2 = Punto2D(5,7)
179 p3 = Punto2D(11,7)
180 p4 = Punto2D(11,0)
181
182 #-----
183 # Definición de las barras
184 #-----
185 # Barra a
186 E_a = 20E6 #20 kN/m^2
187 Aa = 0.1**2 # Sección cuadrada de 0.1 m de lado
188 Ia = 1./12 * 0.1**4
189
190 EA_a = E_a * Aa
191 EI_a = E_a * Ia
192
193 barra_a = P2D(p1,p2,EA_a, EI_a)
194
195 # Barra b
196
197 E_b = 25E6 # 25 kN/m^2
198 Ab = 0.2**2 # Sección cuadrada de 0.2 m de lado
199 Ib = 1./12 * 0.2**4
200
201 EA_b = E_b * Ab
202 EI_b = E_b * Ib
203
204 barra_b = P2D(p2,p3,EA_b, EI_b)
205
206 # Barra c
207 E_c = 200E6 # 200 kN/m^2
208 Ac = np.pi * 0.05**2 / 4. # Barra circular de 5 cm de diámetro
209
210 EA_c = E_c * Ac
211
212 barra_c = A2D(p2,p4,EA_c)
213
214
215 #-----
216 # Determinación de las submatrices
217 #-----
218 # De forma simbólica, es posible eliminar el bloque
219 # correspondiente al nudo 3, pues está empotrado.
220 # También es posible eliminar el bloque del nudo 4, pues están
221 # impedidos los desplazamientos en los 2 gdl que tiene la
222 # barra A2D.
223 # Del sistema restante, almacenaremos en memoria las submatrices
224 # necesarias.
225
226 # Barra a
227 # La barra "a" conecta los nudos 1 y 2, por tanto las posiciones
228 # 11, 12 y 22, se encuentran en las posiciones 11, 12, y 22
229 # de la matriz de rigidez en globales

```

4. Cargas aplicadas en barras y otras acciones

```
224 k11_a = barra_a.K_global_caja(1,1)
225 k12_a = barra_a.K_global_caja(1,2)
226 k22_a = barra_a.K_global_caja(2,2)
227
228
229 # Barra b
230 # La barra "b" conecta los nudos 2 y 3. La submatriz k22_b es la
    única necesaria, y estará en la posición local 11 de la
    matriz de rigidez de la barra, en globales.
231 k22_b = barra_b.K_global_caja(1,1)
232
233
234 # Barra c
235 # La barra "c" conecta los nudos 2 y 4. Se necesita la submatriz
    k22_c, que se encuentran en la posición 11 de la matriz de
    rigidez en globales
236 k22_c = barra_c.K_global_caja(1,1)
237
238 # La barra "c" es un tirante, y necesita expandirse a una 3x3
    con ceros para poderla combinar con las matrices de P2D
239 # k22_c debe expandirse a una matriz de dimensión 3x3, con la ú
    ltima fila y columna de ceros.
240 k22_c_expandida = np.zeros([3,3]) # Primero se crea una matriz
    con ceros
241 k22_c_expandida[0:2,0:2] = k22_c # Luego se copian las
    posiciones de k22_c
242
243
244 # Suma de submatrices. Sólo lo necesita la posición k22_{a+b+c}
245 k22_abc = k22_a + k22_b + k22_c_expandida
246
247 #-----
248 # Ensamblaje
249 #-----
250 # La matriz de rigidez completa tiene dimensiones 6x6:
251 krigidez = np.zeros([6,6])
252
253 krigidez[0:3,0:3] = k11_a
254 krigidez[0:3,3:6] = k12_a
255 krigidez[3:6,0:3] = k12_a.T # Matriz transpuesta de k12_a
256 krigidez[3:6,3:6] = k22_abc
257
258 print("")
259 print("Matriz de rigidez")
260 print(krigidez)
261
262 # En el nudo 1 hay un apoyo deslizante, que impide los
    desplazamientos en los gdl 2 y 3, por lo que es necesario
    eliminar las filas y columnas 2 y 3 de la matriz de rigidez
263 # Para Python, estas filas son las 1 y 2, ya que Python comienza
    indexando en 0
264 # Esto puede hacerse con la operación de numpy "delete"
265
266 posiciones = np.array([2,3]) # Posiciones indexando desde 1
267 posiciones -= 1 # Posiciones de Python, indexando desde 0
268 # Esta operación equivale a posiciones = posiciones - [1,1]
```

```

269 # Primero las filas (axis = 0):
270 krigidez_final = np.delete(krigidez,posiciones,axis = 0)
271 # Luego, las columnas(axis=1):
272 krigidez_final = np.delete(krigidez_final,posiciones,axis = 1)
273
274 # Mostramos la nueva matriz de rigidez:
275 print("")
276 print("Matriz de rigidez tras eliminar filas-columnas 2 y 3")
277 print(krigidez_final)
278
279
280 #-----
281 # Vector de fuerzas de empotramiento
282 #-----
283 # Aparecen fuerzas de empotramiento en los nudos 2 y 3 debido a
284 # la carga distribuida en la barra b.
285 # Sólo interviene la fuerza de empotramiento del nudo 2
286
287 # Calculamos el vector de fuerzas de empotramiento del nudo 2,
288 # debido a la carga q:
289 q = 100 # kN/m
290 Lb = longitud(p2,p3)
291 p2 = np.array([0, q * Lb / 2, q * Lb**2 / 12.])
292
293 vec_fuerzas = np.zeros(6)
294 vec_fuerzas[3:6] = p2
295 vec_fuerzas = np.delete(vec_fuerzas,posiciones)
296
297 # Cambio de signo para tener en cuenta que el sistema es K * u =
298 # F - F_empotramiento:
299 vec_fuerzas = -vec_fuerzas
300
301 #-----
302 ## Resolución del sistema de ecuaciones
303 #-----
304 u = np.linalg.solve(krigidez_final,vec_fuerzas)
305 print("")
306 print("Solución del sistema. Desplazamientos: ")
307 print("u_1x =", u[0], "m")
308 print("u_2x =", u[1], "m")
309 print("u_2y =", u[2], "m")
310 print("Theta_2 =", u[3], "rad")

```

Los resultados de desplazamientos que se obtienen ejecutando este código con los datos indicados son:

$$\begin{aligned}
 u_{1x} &= -1.588 \cdot 10^{-2} \text{ m}; & u_{2x} &= -1.182 \cdot 10^{-3} \text{ m}; \\
 u_{2y} &= -1.037 \cdot 10^{-2} \text{ m}; & \theta_2^{ab} &= -1.281 \cdot 10^{-1} \text{ rad}
 \end{aligned}$$

Capítulo 5

Técnicas complementarias y casos particulares

Índice

5.1. Introducción	89
5.2. Condensación estática y libertades en elementos	90
5.3. Apoyos no concordantes	111
5.4. Apoyos elásticos	114
5.5. Ligaduras entre grados de libertad	117
5.6. Subestructuras	124
5.7. Elementos especiales	125

5.1. Introducción

Con las técnicas estudiadas hasta este punto puede resolverse una gran variedad de estructuras, pero es posible que se presenten algunos casos particulares en los cuales es necesario el conocimiento de ciertas técnicas complementarias para permitir o facilitar la resolución del problema. Algunos de estos casos son:

- Elementos que tienen liberados ciertos gd , por lo que se puede emplear la técnica de condensación estática para simplificar el problema.
- Apoyos orientados en direcciones no concordantes con los ejes globales, donde habrá que definir unos ejes nodales.
- Apoyos elásticos, en los que los desplazamientos son proporcionales a la fuerza transmitida (muelle).

- Presencia de elementos de gran rigidez (sólidos rígidos) que hacen que no todos los **gdl** de la estructura sean independientes, sino que algunos estén ligados entre sí.
- Estructuras de cierto tamaño en las que se repite un mismo patrón, por lo que pueden ser consideradas como un conjunto de subestructuras, facilitando el cálculo.
- Elementos especiales, distintos de las barras o tirantes simples que hemos visto hasta el momento.

En el presente capítulo se van a explicar las técnicas mediante las cuales es posible abordar cada una de estas situaciones particulares.

5.2. Condensación estática y libertades en elementos

Condensación estática

La condensación estática es una técnica numérica utilizada para **eliminar gdl**s del sistema de ecuaciones, despejando unos de ellos (esclavos), en función de otros (maestros), que son los que aparecerán en el sistema reducido. Utilizaremos el superíndice q para hacer referencia a los elementos del sistema condensado.

$$\begin{array}{ccc} \mathbf{F} - \tilde{\mathbf{F}} = \mathbf{K}\mathbf{u} & \xrightarrow{\text{condensación}} & \mathbf{F}^q - \tilde{\mathbf{F}}^q = \mathbf{K}^q\mathbf{u}^q \\ m \text{ ecuaciones} & & n \text{ ecuaciones} \end{array} \quad (5.1)$$

Se pasa de m **gdl**s a n , con $n < m$.

Veamos cómo realizar esta condensación estática planteando un sistema de ecuaciones general para una estructura, donde llamamos A a los **gdl** que queremos mantener (maestros) y B a los **gdl**s a condensar (esclavos):

$$\begin{pmatrix} \mathbf{F}_A \\ \mathbf{F}_B \end{pmatrix} - \begin{pmatrix} \tilde{\mathbf{F}}_A \\ \tilde{\mathbf{F}}_B \end{pmatrix} = \begin{pmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{BA} & \mathbf{K}_{BB} \end{pmatrix} \begin{pmatrix} \mathbf{u}_A \\ \mathbf{u}_B \end{pmatrix}$$

Tomamos las filas de los **gdl** a condensar:

$$\mathbf{F}_B - \tilde{\mathbf{F}}_B = \mathbf{K}_{BA}\mathbf{u}_A + \mathbf{K}_{BB}\mathbf{u}_B$$

y despejando \mathbf{u}_B :

$$\mathbf{u}_B = \mathbf{K}_{BB}^{-1}\mathbf{F}_B - \mathbf{K}_{BB}^{-1}\tilde{\mathbf{F}}_B - \mathbf{K}_{BB}^{-1}\mathbf{K}_{BA}\mathbf{u}_A \quad (5.2)$$

Tomamos ahora las filas de los **gdl** restantes:

$$\mathbf{F}_A - \tilde{\mathbf{F}}_A = \mathbf{K}_{AA}\mathbf{u}_A + \mathbf{K}_{AB}\mathbf{u}_B$$

Sustituyendo la Ecuación 5.2 en la Ecuación 5.3:

$$\mathbf{F}_A - \tilde{\mathbf{F}}_A = \mathbf{K}_{AA}\mathbf{u}_A + \mathbf{K}_{AB}(\mathbf{K}_{BB}^{-1}\mathbf{F}_B - \mathbf{K}_{BB}^{-1}\tilde{\mathbf{F}}_B - \mathbf{K}_{BB}^{-1}\mathbf{K}_{BA}\mathbf{u}_A)$$

y agrupando términos,

$$(\mathbf{F}_A - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\mathbf{F}_B) - (\tilde{\mathbf{F}}_A - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\tilde{\mathbf{F}}_B) = (\mathbf{K}_{AA} - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{BA})\mathbf{u}_A \quad (5.3)$$

Vemos que los gdl B ya no aparecen en el sistema. En esta expresión podemos identificar los términos del sistema condensado indicados en la Ecuación 5.1:

$$\begin{aligned} \mathbf{F}^q &= \mathbf{F}_A - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\mathbf{F}_B \\ \tilde{\mathbf{F}}^q &= \tilde{\mathbf{F}}_A - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\tilde{\mathbf{F}}_B \\ \mathbf{K}^q &= \mathbf{K}_{AA} - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{BA} \\ \mathbf{u}^q &= \mathbf{u}_A \end{aligned} \quad (5.4)$$

Esta será la forma de obtener los elementos del sistema de ecuaciones condensado. A modo de resumen tenemos:

- A : gdl que permanecen en el sistema de ecuaciones (*gdl maestros*).
- B : gdl que se eliminan del sistema de ecuaciones (*gdl esclavos*).
- \mathbf{F}_A y $\tilde{\mathbf{F}}_A$: fuerzas existentes en los gdl a mantener.
- \mathbf{F}_B y $\tilde{\mathbf{F}}_B$: fuerzas existentes en los gdl a condensar.
- \mathbf{u}_A : desplazamientos en los gdl a mantener.
- \mathbf{u}_B : desplazamientos en los gdl a condensar.
- \mathbf{K}_{AA} : términos de la matriz de rigidez que relacionan fuerzas y desplazamientos a mantener.
- \mathbf{K}_{BB} : términos de la matriz de rigidez que relacionan fuerzas y desplazamientos a condensar.
- \mathbf{K}_{AB} : términos de la matriz de rigidez que relacionan fuerzas a mantener con desplazamientos a condensar.
- \mathbf{K}_{BA} : términos de la matriz de rigidez que relacionan fuerzas a condensar con desplazamientos a mantener.

Vamos a verlo de forma práctica con un ejemplo .

Ejercicio 5.1 *Dada la siguiente barra de pórtico bidimensional sometida a carga uniformemente distribuida:*

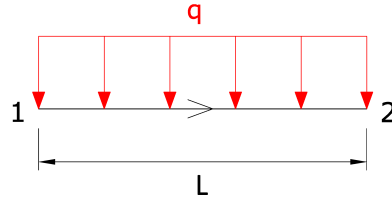


Figura 5.1: Ejercicio 5.1. Barra P2D con carga uniforme.

Calcular su sistema de ecuaciones condensando el giro en el nudo 2.

Solución

La barra tiene 6 *gdl* antes de realizar la condensación, que pasarán a ser 5 tras condensar el sexto *gdl* (giro en el nudo 2):

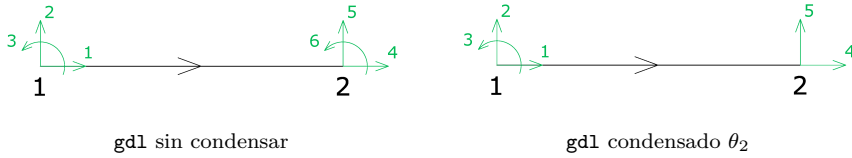


Figura 5.2: Ejercicio 5.1. *gdl* de la barra antes y después de condensar θ_2 .

Planteamos el sistema de ecuaciones $\mathbf{F} - \tilde{\mathbf{F}} = \mathbf{K}\mathbf{u}$ sin condensar. Se muestran calculados los términos de la matriz de rigidez elemental para P2D y las fuerzas de empotramiento correspondientes a una carga distribuida uniforme. Se han incluido unas fuerzas nodales genéricas en ambos nudos de la barra, para mayor generalidad del ejemplo.

$$\begin{pmatrix} F_{1x} \\ F_{1y} \\ M_1 \\ F_{2x} \\ F_{2y} \\ M_2 \end{pmatrix} - \begin{pmatrix} 0 \\ \frac{qL}{2} \\ \frac{qL^2}{12} \\ 0 \\ \frac{qL}{2} \\ -\frac{qL^2}{12} \end{pmatrix} = \begin{pmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^3} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^3} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^3} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^3} & \frac{4EI}{L} \end{pmatrix} \begin{pmatrix} u_{1x} \\ u_{1y} \\ \theta_1 \\ u_{2x} \\ u_{2y} \\ \theta_2 \end{pmatrix}$$

Como se ha indicado, el *gdl* a condensar es el 6, por lo que se han señalado la sexta fila del sistema y la sexta columna de la matriz de rigidez,

lo que nos permitirá identificar los diferentes términos para la condensación:

$$\mathbf{F}_A = \begin{pmatrix} F_{1x} \\ F_{1y} \\ M_1 \\ F_{2x} \\ F_{2y} \end{pmatrix}; \quad \mathbf{F}_B = (M_2); \quad \tilde{\mathbf{F}}_A = \begin{pmatrix} 0 \\ \frac{qL}{2} \\ \frac{qL^2}{12} \\ 0 \\ \frac{qL}{2} \end{pmatrix}; \quad \tilde{\mathbf{F}}_B = \left(-\frac{qL^2}{12}\right)$$

$$\mathbf{K}_{AA} = \begin{pmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} \end{pmatrix}; \quad \mathbf{K}_{AB} = \begin{pmatrix} 0 \\ \frac{6EI}{L^2} \\ \frac{2EI}{L} \\ 0 \\ -\frac{6EI}{L^2} \end{pmatrix}$$

$$\mathbf{K}_{BA} = \left(0 \quad \frac{6EI}{L^3} \quad \frac{2EI}{L} \quad 0 \quad -\frac{6EI}{L^3}\right); \quad \mathbf{K}_{BB} = \left(\frac{4EI}{L}\right)$$

$$\mathbf{u}_A = \begin{pmatrix} u_{1x} \\ u_{1y} \\ \theta_1 \\ u_{2x} \\ u_{2y} \end{pmatrix}; \quad \mathbf{u}_B = (\theta_2)$$

Calculamos ahora los elementos del sistema condensado, según las expresiones vistas en el presente apartado.

- Vector de fuerzas condensado.

Como norma, condensaremos un determinado *gdl* únicamente cuando no haya fuerzas aplicadas en el mismo ($\mathbf{F}_B = 0$), lo que en este caso implica $M_2 = 0$. Esto lo hacemos para poder escribir que:

$$\mathbf{F}^q = \mathbf{F}_A - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\mathbf{F}_B \Rightarrow \mathbf{F}^q = \mathbf{F}_A$$

- Vector de fuerzas de empotramiento condensado.

$$\begin{aligned}\tilde{\mathbf{F}}^q &= \tilde{\mathbf{F}}_A - \mathbf{K}_{AB} \mathbf{K}_{BB}^{-1} \tilde{\mathbf{F}}_B \Rightarrow \\ \Rightarrow \tilde{\mathbf{F}}^q &= \begin{pmatrix} 0 \\ \frac{qL}{2} \\ \frac{qL^2}{12} \\ 0 \\ \frac{qL}{2} \end{pmatrix} - \begin{pmatrix} 0 \\ \frac{6EI}{L^2} \\ \frac{2EI}{L} \\ 0 \\ -\frac{6EI}{L^2} \end{pmatrix} \begin{pmatrix} L \\ 4EI \end{pmatrix} \begin{pmatrix} -\frac{qL^2}{12} \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{5qL}{8} \\ \frac{qL^2}{8} \\ 0 \\ \frac{3qL}{8} \end{pmatrix}\end{aligned}$$

Se puede observar que las fuerzas obtenidas corresponden a las reacciones de una barra empotrada-apoyada (barra con el *gdl* 6 liberado) sometida a la carga uniforme q . Esto nos permitirá, de aquí en adelante, calcular de forma directa las fuerzas de empotramiento cuando condensemos un *gdl*.

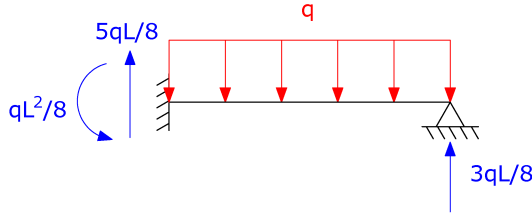


Figura 5.3: Ejercicio 5.1. Reacciones en barra empotrada-apoyada con carga uniforme.

- Matriz de rigidez condensada.

$$\mathbf{K}^q = \mathbf{K}_{AA} - \mathbf{K}_{AB} \mathbf{K}_{BB}^{-1} \mathbf{K}_{BA}$$

Sustituyendo y operando:

$$\mathbf{K}^q = \begin{pmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 \\ 0 & \frac{3EI}{L^3} & \frac{3EI}{L^2} & 0 & -\frac{3EI}{L^3} \\ 0 & \frac{3EI}{L^2} & \frac{3EI}{L} & 0 & -\frac{3EI}{L^2} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 \\ 0 & -\frac{3EI}{L^3} & -\frac{3EI}{L^2} & 0 & \frac{3EI}{L^2} \end{pmatrix}$$

- Vector de desplazamientos condensado.

$$\mathbf{u}^q = \mathbf{u}_A$$

En total, el sistema de ecuaciones condensado queda:

$$\begin{pmatrix} F_{1x} \\ F_{1y} \\ M_1 \\ F_{2x} \\ F_{2y} \end{pmatrix} - \begin{pmatrix} 0 \\ \frac{5qL}{8} \\ \frac{qL^2}{8} \\ 0 \\ \frac{3qL}{8} \end{pmatrix} = \begin{pmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 \\ 0 & \frac{3EI}{L^3} & \frac{3EI}{L^2} & 0 & -\frac{3EI}{L^3} \\ 0 & \frac{3EI}{L^2} & \frac{3EI}{L} & 0 & -\frac{3EI}{L^2} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 \\ 0 & -\frac{3EI}{L^3} & -\frac{3EI}{L^2} & 0 & \frac{3EI}{L^2} \end{pmatrix} \begin{pmatrix} u_{1x} \\ u_{1y} \\ \theta_1 \\ u_{2x} \\ u_{2y} \end{pmatrix}$$

Libertades en elementos

En el Ejercicio 5.1 hemos condensado, a modo de ejemplo, el giro en el extremo final de una barra cualquiera. Aunque este procedimiento se puede llevar a cabo para cualquier **gdl**, la verdadera utilidad de la condensación estática se muestra cuando nos encontramos con libertades en elementos.

Hablamos de libertad en un elemento cuando alguno de sus nudos tiene liberado al menos un **gdl**, es decir, permite el desplazamiento en dicho **gdl** (mediante una rótula o una deslizadera). Esta situación se puede dar en nudos con apoyos o en nudos entre barras, en cuyo caso tendremos un nudo con más **gdl** que los habituales en una estructura de ese tipo. La mejor forma de resolver estas situaciones es condensando los **gdl** liberados.

Para la resolución de problemas, aplicaremos la condensación estática siempre que se den las siguientes circunstancias simultáneamente:

- Cuando esté **liberado el giro o el desplazamiento perpendicular** a la barra en uno de sus extremos. Otros casos de condensación son también factibles, pero estos son los más comunes y es posible encontrar las matrices de rigidez condensadas en prontuarios de Cálculo Matricial, como el del Anexo B del presente libro.
- Cuando **no existan cargas externas** aplicadas en el **gdl** a condensar, es decir, la componente de dicho **gdl** en el vector \mathbf{F} sea igual a cero. Esto permite obtener de forma directa el vector de fuerzas condensado como $\mathbf{F}^q = \mathbf{F}_A$.

Veamos algunos ejemplos de situaciones en las que es conveniente condensar **gdl**:

- Ejemplo 1. Barras conectadas por una rótula.

En la rótula tendremos 4 **gdl**: desplazamiento horizontal, desplazamiento vertical, giro por la izquierda y giro por la derecha, ya que la rótula hace que el giro sea distinto por ambos lados. Estos dos giros,

que están liberados por la presencia de la rótula, se pueden y deben condensar:

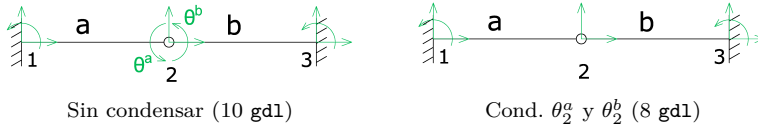


Figura 5.4: Condensación en barras unidas por rótula.

Hay que tener en cuenta que en el caso de que haya fuerzas nodales en un determinado gd1, no lo condensaremos, como se ha indicado anteriormente. Si, por ejemplo, en el caso anterior añadimos un momento en la rótula por la derecha, podremos condensar tan solo el giro en el nudo 2 por la izquierda:

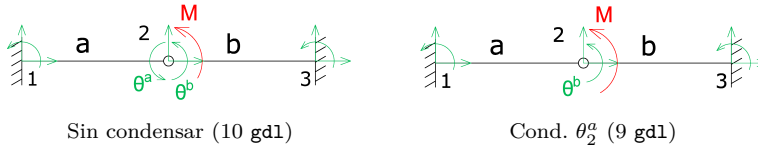


Figura 5.5: Condensación en barras unidas por rótula con momento en nudo.

■ Ejemplo 2. Barras conectadas por una deslizadera.

En la figura siguiente, la deslizadera independiza el desplazamiento vertical en el nudo 2 de la barra a , por un lado, y de las barras b y c , por otro. Se podrá condensar el desplazamiento u_{2y}^a , ya que está libre y es perpendicular a la barra a en uno de sus extremos.

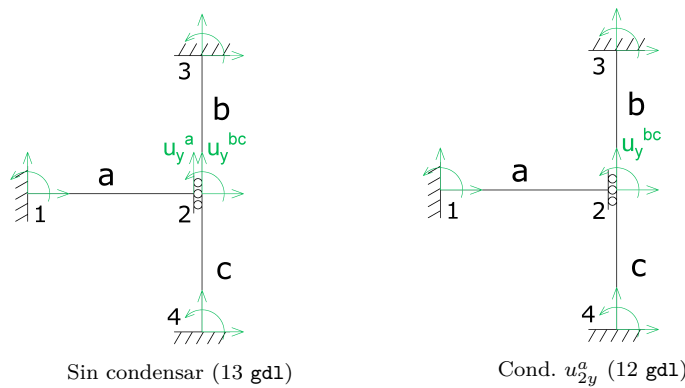


Figura 5.6: Condensación en barras unidas con deslizadera

■ Ejemplo 3. Pórtico con rótula interna y apoyos no empotrados.

En el pórtico a continuación, se podrán condensar los giros en el nudo 2 (liberados por la rótula) y el desplazamiento en la deslizadera del nudo 4, que es perpendicular a la barra c .

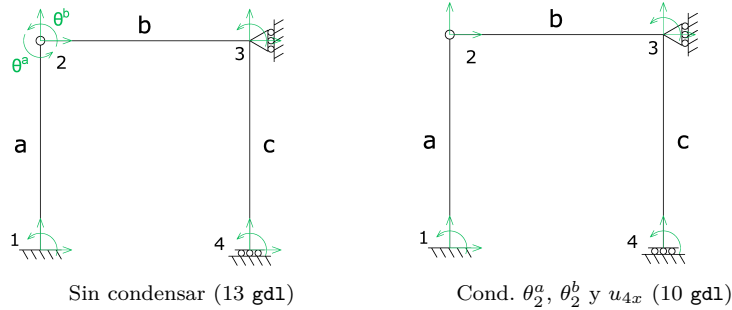


Figura 5.7: Ejemplo de condensación en pórtico

■ Ejemplo 5. Pórtico con rótula interna.

En la figura siguiente, las barras a y b están rígidamente unidas entre sí en el nudo 3, por lo que no puede condensarse su giro. Las barras c y d , sin embargo, tienen los giros libres en ese nudo, por lo que deberán condensarse. El desplazamiento en la deslizadera del nudo 1 no se condensa porque no es perpendicular al eje de la barra a .

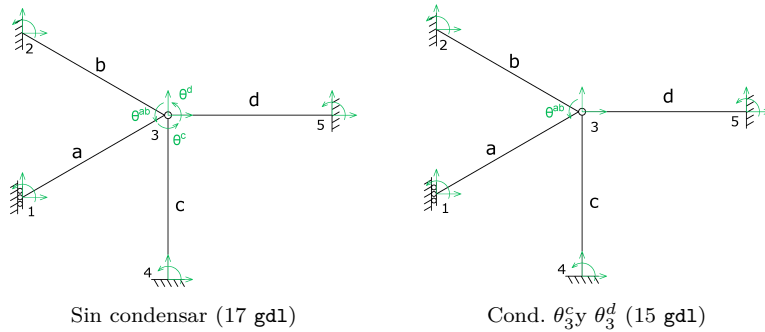


Figura 5.8: Ejemplo de condensación en pórtico (II)

La aplicación de la condensación estática en gdl de elementos que coinciden en un nudo debe hacerse teniendo precaución de no hacer inestable el propio nudo. Si se condensan, por ejemplo, todos los giros del nudo, ninguna barra aportará rigidez de giro al nudo y se producirá un mecanismo interno (a menos que se coaccionara de forma artificial dicho giro). Esto es importante si se va a resolver el problema con un programa de ordenador, pero no

si se va a hacer a mano, pues en este último caso, puede tenerse en cuenta que todos **gdl**s condensados se eliminan del sistema de ecuaciones.

Ejercicio 5.2 Se tiene la siguiente estructura, donde las barras a y b se unen rígidamente entre sí, y la barra c se une a ellas a través de una rótula.

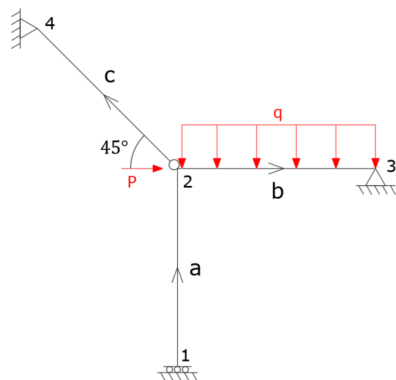


Figura 5.9: Ejercicio 5.2. Estructura.

Se pide calcular los desplazamientos en los nudos 2 y 3 y los esfuerzos en la barra horizontal mediante el Método Directo de la Rigidez y utilizando el mínimo número de **gdl** posibles.

Datos: $EI = 2000 \text{ kNm}^2$; $EA = 1.2 \cdot 10^5 \text{ kN}$; longitud de todas las barras $L = 5 \text{ m}$; $q = 25 \text{ kN/m}$; $P = 100 \text{ kN}$.

Solución

Es posible observar que tenemos una estructura en la que las barras a y b son de tipo **P2D**, mientras que la barra c es de tipo **A2D**. Además, siguiendo los criterios vistos en el presente apartado, podemos condensar el desplazamiento horizontal en el nudo 1 y el giro en el nudo 3, ya que ambos **gdl** están liberados y no tienen cargas nodales aplicadas.

De este modo, definimos los siguientes **gdl** de la estructura:

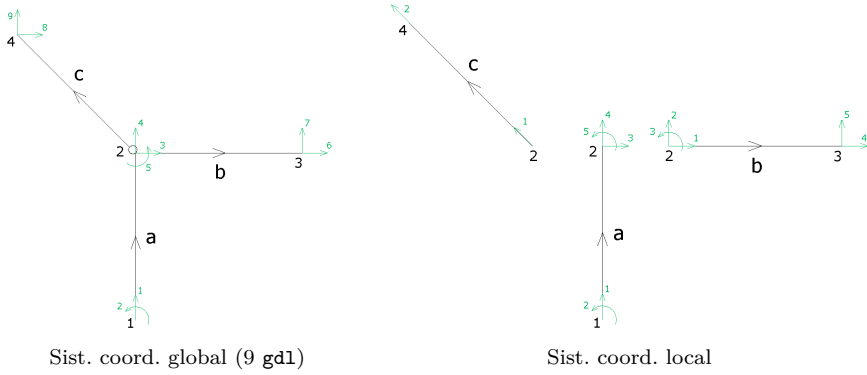


Figura 5.10: Ejercicio 5.2. gdl de la estructura condensados u_{1x} y θ_3 .

Planteamos el sistema matricial de ecuaciones de la estructura $\mathbf{F} - \tilde{\mathbf{F}} = \mathbf{K}\mathbf{u}$ teniendo en cuenta los gdl que están condensados:

$$\begin{pmatrix} R_{1y} \\ M_1 \\ P \\ 0 \\ 0 \\ R_{3x} \\ R_{3y} \\ R_{4x} \\ R_{4y} \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \tilde{\mathbf{P}}_2^b \\ \tilde{\mathbf{P}}_3^b \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} k_{11}^a & k_{12}^a & 0 & 0 \\ k_{21}^a & k_{22}^{a+b+c} & k_{23}^b & k_{24}^c \\ 0 & k_{32}^b & k_{33}^b & 0 \\ 0 & k_{42}^c & 0 & k_{44}^c \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ u_{2x} \\ u_{2y} \\ \theta_2^{ab} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Particionamos el sistema seleccionando las ecuaciones que tienen a los desplazamientos como incógnitas (ecuaciones de la tercera a la quinta, correspondientes al nudo 2). En este caso, el sistema a resolver queda:

$$\mathbf{F}_2 - \tilde{\mathbf{P}}_2^b = \mathbf{k}_{22}^{a+b+c} \mathbf{u}_2$$

Calculamos las fuerzas de empotramiento en el nudo 2 producidas por la carga distribuida q aplicada sobre la barra b. Se trata del mismo caso que el descrito en el Ejercicio 5.1, basta calcular las reacciones de una barra empotrada-apoyada sometida a una carga uniformemente distribuida.

$$\tilde{\mathbf{P}}_2^b = \begin{pmatrix} 0 \\ \frac{5qL}{8} \\ \frac{qL^2}{8} \end{pmatrix} = \begin{pmatrix} 0 \\ 78.13 \\ 78.13 \end{pmatrix}$$

Calculamos ahora los términos de \mathbf{K} que intervienen en el sistema, es decir, los correspondientes a la posición 22:

- Barra a (P2D; $L = 5$ m; $\alpha = 90^\circ$; condensado u_{1x})

$$\mathbf{k}^a = \left(\begin{array}{c|c} \mathbf{k}_{11}^a & \mathbf{k}_{12}^a \\ \hline \mathbf{k}_{21}^a & \mathbf{k}_{22}^a \end{array} \right) \Rightarrow \mathbf{k}_{22}^a = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 24000 & 0 \\ 0 & 0 & 400 \end{pmatrix}$$

- Barra b (P2D; $L = 5$ m; $\alpha = 0^\circ$; condensado θ_3)

$$\mathbf{k}^b = \left(\begin{array}{c|c} \mathbf{k}_{22}^b & \mathbf{k}_{23}^b \\ \hline \mathbf{k}_{32}^b & \mathbf{k}_{33}^b \end{array} \right) \Rightarrow \mathbf{k}_{22}^b = \begin{pmatrix} 24000 & 0 & 0 \\ 0 & 48 & 240 \\ 0 & 240 & 1200 \end{pmatrix}$$

- Barra c (A2D; $L = 5$ m; $\alpha = 135^\circ$)

$$\mathbf{k}^c = \left(\begin{array}{c|c} \mathbf{k}_{22}^c & \mathbf{k}_{24}^c \\ \hline \mathbf{k}_{42}^c & \mathbf{k}_{44}^c \end{array} \right) \Rightarrow \mathbf{k}_{22}^c = \begin{pmatrix} 12000 & -12000 & 0 \\ -12000 & 12000 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Sumando las matrices podemos plantear el sistema de ecuaciones a resolver y calcular los desplazamientos:

$$\begin{pmatrix} 100 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 78.13 \\ 78.13 \end{pmatrix} = \begin{pmatrix} 36000 & -12000 & 0 \\ -12000 & 36048 & 240 \\ 0 & 240 & 1600 \end{pmatrix} \begin{pmatrix} u_{2x} \\ u_{2y} \\ \theta_2^{ab} \end{pmatrix}$$

Se obtiene:

$$u_{2x} = 2.433 \cdot 10^{-3} \text{ m}; \quad u_{2y} = -1.033 \cdot 10^{-3} \text{ m}; \quad \theta_2 = -4.867 \cdot 10^{-2} \text{ rad}$$

Obtenidos estos desplazamientos es posible calcular los esfuerzos en la barra b.

$$\begin{pmatrix} \mathbf{P}_2^b \\ \mathbf{P}_3^b \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{P}}_2^b \\ \tilde{\mathbf{P}}_3^b \end{pmatrix} + \left(\begin{array}{c|c} \mathbf{k}_{22}^b & \mathbf{k}_{23}^b \\ \hline \mathbf{k}_{32}^b & \mathbf{k}_{33}^b \end{array} \right) \begin{pmatrix} \mathbf{u}_2 \\ \mathbf{u}_3 \end{pmatrix} \Rightarrow \mathbf{P}_2^b = \tilde{\mathbf{P}}_2^b + \mathbf{k}_{22}^b \mathbf{u}_2 = \begin{pmatrix} 58.39 \\ 66.39 \\ 19.77 \end{pmatrix}$$

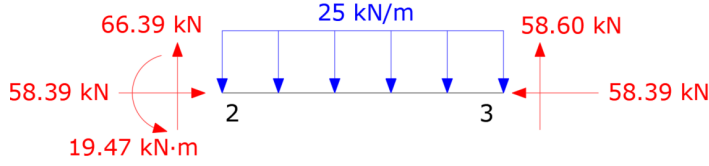


Figura 5.11: Ejercicio 5.2. Fuerzas la barra b.

Dando lugar a los siguientes diagramas de esfuerzos:

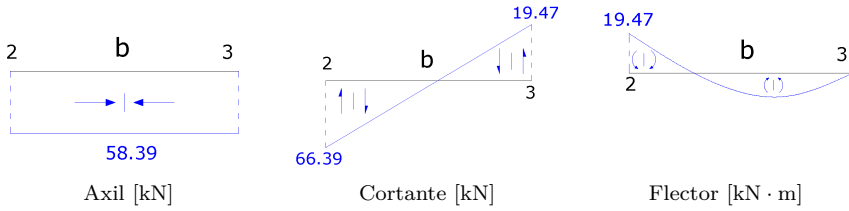


Figura 5.12: Ejercicio 5.2. Diagramas de esfuerzos en la barra b.

Se pide también calcular el giro en el nudo 3, que está condensado. Para obtenerlo tendremos que trabajar con la ecuación de la barra b sin condensar:

$$\begin{pmatrix} \bar{P}_2^b \\ \bar{P}_3^b \end{pmatrix} - \begin{pmatrix} \tilde{P}_2^b \\ \tilde{P}_3^b \end{pmatrix} = \begin{pmatrix} k_{22}^b & k_{23}^b \\ k_{32}^b & k_{33}^b \end{pmatrix} \begin{pmatrix} u_2 \\ u_3 \end{pmatrix}$$

Bastará con resolver la sexta fila de este sistema, correspondiente al gdl buscado. Calculamos sólo los términos de las matrices y vectores que necesitamos:

- *Fuerzas de empotramiento.*

Estamos trabajando ahora con la barra sin condensar ningún gdl , así que debemos calcular las fuerzas de empotramiento como las reacciones en una barra biempotrada sometida a una carga uniformemente distribuida.

$$\tilde{\mathbf{P}}_3^b = \begin{pmatrix} 0 \\ \frac{qL}{2} \\ -\frac{qL^2}{12} \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{qL}{2} \\ -52.08 \end{pmatrix}$$

- *Términos de \mathbf{k}^b (sin condensar).*

$$\mathbf{k}_{32}^b = \begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ 0 & 480 & 800 \end{pmatrix}; \quad \mathbf{k}_{33}^b = \begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & 1600 \end{pmatrix}$$

$$\mathbf{u}_2 = \begin{pmatrix} 2.433 \cdot 10^{-3} \\ -1.033 \cdot 10^{-3} \\ -4.867 \cdot 10^{-2} \end{pmatrix}; \quad \mathbf{u}_3 = \begin{pmatrix} 0 \\ 0 \\ \theta_3 \end{pmatrix}$$

$$0 + 52.08 = 480(-1.033 \cdot 10^{-3}) + 800(-4.867 \cdot 10^{-2}) + 1600\theta_3 \Rightarrow$$
$$\Rightarrow \theta_3 = 5.720 \cdot 10^{-2} \text{ rad}$$

Ejercicio Python 5.1

The diagram shows a continuous beam of total length $4L_1$. It has a pin support at the left end (point 1) and a roller support at the right end (point 4). There are two internal hinges at points 2 and 3. A uniformly distributed load q acts downwards over the entire length of the beam. The beam is divided into three segments: a (length L_1), b (length L_2), and c (length L_1). At point 2, there is a vertical support of height h_1 and a horizontal distance d_1 from the hinge to the support. At point 3, there is a vertical support of height h_2 and a horizontal distance d_2 from the hinge to the support. The beam is labeled with points 1, 2, 3, 4, 5, and 6. Points 5 and 6 are the supports for the vertical supports at points 2 and 3, respectively.

Se pide plantear y resolver el sistema de ecuaciones para calcular los desplazamientos condensando todos los **gdl** posibles.

- Tablero: $E = 35 \text{ GPa}$; $A = 3.5 \text{ m}^2$; $I = 0.12 \text{ m}^4$.
- Pilas: $E = 35 \text{ GPa}$; $A = 0.2 \text{ m}^2$; $I = 5.3 \cdot 10^{-3} \text{ m}^4$.

- Dimensiones: $L1 = 30$ m; $L2 = 45$ m; $h1 = h2 = 20$ m; $d1 = d2 = 5$ m.

El código para resolver este problema, a continuación, comienza con la definición de las clases “Punto2D”, “P2D” y “A2D”, con sus correspondientes funciones, ya utilizadas en el ejercicio de Python 4.1. Además, se añade a la clase “P2D” una función “K_Condensa_Giro” que obtiene la matriz de rigidez de una barra P2D con el giro condensado en uno de los dos extremos.

Una vez definidas las funciones, se sigue el siguiente procedimiento para resolver el problema:

1. Definir la geometría de la estructura y la posición de los nudos de la estructura con la clase “Punto2D”. Estos puntos se muestran en pantalla.
2. Definir las barras introduciendo los materiales y usando la clase “P2D”.
3. Se realiza un dibujo sencillo de la estructura para visualizar su geometría.
4. Calcular las submatrices necesarias y ensamblar la matriz del sistema teniendo en cuenta las condiciones de contorno y los \mathbf{gdl} condensados $(\theta_1, \theta_2^d, \theta_3^e, \theta_4)$. De este modo el sistema a resolver queda de dimensiones 7×7
5. Definir el vector de fuerzas de empotramiento producidas por la carga distribuida q .
6. Resolver el sistema de ecuaciones para calcular los desplazamientos.

```

1 from __future__ import division
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 #=====
6 # DEFINICIÓN DE CLASES Y FUNCIONES
7 # Matrices de Rigidez P2D y A2D en globales
8 # Completa y con submatrices
9 # Programación Orientada a Objetos
10 #=====
11
12 #-----
13 # Definición de la clase "Punto". Coordenadas (x, y) de un punto
14 # Observación: Se inicializa con las coordenadas (0,0)
15 # Se define bidimensional, pues se usará para el P2D
16 #-----
17 class Punto2D:
18     """ Clase para representar los puntos en 3D, coordenadas x,
19     y, z """
20     def __init__(self, x=0, y=0):
21         self.x = x
22         self.y = y
23     def coords(self):

```

```

23         return "{0}, {1}".format(self.x, self.y)
24
25 #-----
26 # Definición de la función "longitud"
27 # Calcula la distancia entre dos puntos
28 # Trabaja con p1,p2. Es una función "amiga" de la clase Punto
29 #-----
30 def longitud(p1, p2):
31     """ Devuelve la longitud (módulo) del vector p1-p2 """
32     Longitud = 0
33     Longitud += (p2.x - p1.x ) ** 2
34     Longitud += (p2.y - p1.y ) ** 2
35     Longitud = np.sqrt(Longitud)
36     return Longitud
37
38 #-----
39 # Definición de la función "alfa"
40 # Devuelve el ángulo que define el vector P1 -> P2
41 # La función "arctan2" devuelve el ángulo buscado
42 #-----
43 def alfa(p1,p2):
44     """Angulo P1 -> P2"""
45     vy = p2.y -p1.y
46     vx = p2.x -p1.x
47     angulo = np.arctan2(vy,vx)
48     return angulo
49
50 #-----
51 # Definición de la clase "P2D", para barra P2D
52 #-----
53 # Inicialización: A partir de dos puntos, y los parámetros EA,
54 #   EI
55 # Funciones de la clase:
56 #   K_local: Para obtener la matriz de rigidez en locales (6x6
57 #   )
58 #   M_lg: Matriz de cambio de base de Locales a Globales (6x6)
59 #   K_global: Para obtener la matriz de rigidez en globales (6
60 #   x6)
61 #   K_global_caja(i,j): Para obtener la submatriz ij en
62 #   globales (3x3)
63
64 class P2D:
65     """ Clase para representar la barra P2D """
66     def __init__(self, p1 = Punto2D(), p2 = Punto2D(), ea = 0,
67         ei = 0):
68         self.p1 = p1
69         self.p2 = p2
70         self.ea = ea
71         self.ei = ei
72
73     def K_local(self):
74         L = longitud(self.p1,self.p2)
75         EA = self.ea
76         EI = self.ei
77
78         K_L = np.zeros((6,6))

```

```

74
75     K_L[0,:] = np.array([ EA / L, 0., 0., - EA / L, 0., 0.])
76     K_L[1,1:] = np.array([12 * EI / L**3, 6* EI / L**2, 0,
77                           -12 * EI / L**3, 6 * EI / L**2])
78     K_L[2,2:] = np.array([4 * EI / L, 0, -6 * EI / L**2, 2 *
EI / L ])
79     K_L[3,3:] = np.array([EA / L, 0, 0])
80     K_L[4,4:] = np.array([12 * EI / L**3, -6 * EI / L**2])
81     K_L[5,5] = 4 * EI / L
82
83     K_L = (K_L + K_L.T) - np.diag ( np.diag(K_L) )
84
85     return K_L
86
87 def M_lg(self):
88     Angulo = alfa (self.p1, self.p2)
89
90     M_LG = np.array([[np.cos(Angulo), -np.sin(Angulo), 0],
91                      [np.sin(Angulo), np.cos(Angulo), 0],
92                      [0, 0, 1]])
93     M_LG = np.vstack([np.hstack([M_LG, np.zeros((3,3))]),
94                       np.hstack([np.zeros((3,3)), M_LG])])
95
96     return M_LG
97
98 def K_global(self):
99
100     M_LG = self.M_lg()
101     K_L = self.K_local()
102
103     K_G = np.dot(M_LG, np.dot(K_L , M_LG.T ))
104
105     return K_G
106
107 # Función para extraer una submatriz
108 def K_global_caja(self,icol=1,jcol=1):
109     if (icol not in ([1,2]) or jcol not in ([1,2]) ):
110         print("Valores de icol, jcol fuera de rango")
111         return
112     icol = icol -1
113     jcol = jcol -1
114     kg = self.K_global()
115
116     K_sub = kg[icol * 3:icol * 3 + 3 , jcol * 3 : jcol * 3 +
3]
117
118     return K_sub
119
120
121 # Matriz de rigidez con giro condensado en globales
122 # La operación de condensación se realiza en esta función
123
124 def K_Condensa_Giro(self,nodo=2) :
125     # Si nodo = 1, giro condensado en nodo inicial
126     # Si nodo = 2, giro condensado en nodo final
127     indice = nodo*3-1 # Índice local del GDL en giro, en

```

```

vector de Python
128
129     kg = self.K_global()
130
131 #     k11,k12,k21,k22 son matrices-bloque necesarias para
condensar el giro
132 #     k11: submatriz 5x5. A partir de kg, eliminando F-C del
GDL condensado
133 #     k12: columna 5x1. En la posición del GDL condensado
134 #     k21: fila 1x5. En la posición del GDL condensado.
135 #     k22: componente de la diagonal en el GDL condensado
136     k11 = np.delete(kg,indice,axis=0)
137     k11 = np.delete(k11,indice,axis=1)
138     # Se extrae la columna k12 en formato matriz
139     # Observe cómo se extraen en Python columnas, respetando
el formato columna
140     k12 = kg.T[[indice]].T
141     k12 = np.delete(k12,indice,axis=0) #se elimina la fila "
indice"
142     # Se extrae la fila "indice" en formato matriz
143     k21 = kg[[indice]] # Directamente extrae la fila "indice
" en forma de matriz
144     k21 = np.delete(k21,indice,axis=1) #se elimina la columna
"indice"
145     k22 = kg[indice,indice]
146
147     return k11 - k12.dot(k21) / k22 # k12.dot(k21)=np.dot(
k12,k21)
148
149
150 # -----
151 # Definición de la clase "A2D", para barra A2D
152 # -----
153 # Inicialización: A partir de dos puntos, y el parámetro ea
154 # Funciones de la clase:
155 #     K_local : Para obtener la matriz de rigidez en locales (2
x2)
156 #     M_lg : Matriz de cambio de base de Locales a Globales (4x2
)
157 #     K_global: Para obtener la matriz de rigidez en globales (6
x6)
158 #     K_global_caja(i,j): Para obtener la submatriz en ij en
globales (2x2)
159
160
161 class A2D:
162     """ Clase para representar la barra A2D """
163     def __init__(self, p1 = Punto2D(), p2 = Punto2D(), ea = 0):
164         self.p1 = p1
165         self.p2 = p2
166         self.ea = ea
167
168     def K_local(self):
169         L = longitud(self.p1,self.p2)
170         EA = self.ea
171         K_L = np.array([[EA / L, -EA / L],

```

```

172         [-EA / L, EA / L]])
173     return K_L
174
175     def M_lg(self):
176         Angulo = alfa (self.p1, self.p2)
177         M_LG = np.array([[np.cos(Angulo), 0],
178                         [np.sin(Angulo), 0],
179                         [0, np.cos(Angulo)],
180                         [0, np.sin(Angulo)]])
181         return M_LG
182
183     def K_global(self):
184         M_LG = self.M_lg()
185         K_L = self.K_local()
186         K_G = np.dot(M_LG, np.dot(K_L , M_LG.T ))
187         return K_G
188
189 # Función para extraer una submatriz
190 def K_global_caja(self,icol=1,jcol=1):
191     if (icol not in ([1,2]) or jcol not in ([1,2]) ):
192         print("Valores de icol, jcol fuera de rango")
193     return
194     icol = icol -1
195     jcol = jcol -1
196     kg = self.K_global()
197     K_sub = kg[icol * 2:icol * 2 + 2 , jcol * 2 : jcol * 2 +
2]
198     return K_sub
199
200
201 #=====
202 # RESOLUCIÓN DEL PROBLEMA
203 # Cálculo de estructura P2D - A2D con condensación de gdl
204 #=====
205 #-----
206 # Definición de los parámetros geométricos
207 #-----
208 L1 = 30 #m
209 L2 = 45 #m
210 h1 = 20 #m
211 d1 = 5 #m
212 h2 = 20 #m
213 d2 = 5 #m
214
215 # Definición de los puntos a partir de los parámetros geomé
    tricos
216
217 # Puntos 1,2,3,4, que definen el tablero
218 p1 = Punto2D(0,0)
219 p2 = Punto2D(L1,0)
220 p3 = Punto2D(L1+L2,0)
221 p4 = Punto2D(L1+L2+L1,0)
222
223 # Puntos 5,6, apoyos. Observe el uso de coordenadas relativas a
    los puntos 1 a 4
224 p5 = Punto2D(p2.x-d1, p2.y-h1)

```

```

225 p6 = Punto2D(p3.x+d2, p3.y-h2)
226
227 # Verificación de los puntos. Salida por pantalla
228 listapuntos = [p1,p2,p3,p4,p5,p6]
229
230 print("Listado de puntos")
231 npunto = 1
232 for punto in listapuntos:
233     print("Punto", npunto, "Coordenadas", punto.coords())
234     npunto +=1
235 print("")
236
237 #-----
238 # Definición de los parámetros mecánicos de las barras
239 #-----
240 Elast = 35E6 #Módulo de elasticidad, kN/m2
241 At = 3.5     #Área sección tablero, m2
242 It = 0.12    #Inercia sección tablero, m4
243
244 Ap1 = 0.2    #Área pila 1, m2
245 Ip1 = 5.3E-3 #Inercia pila 1, m4
246
247 Ap2 = 0.2    #Área pila 2, m2
248 Ip2 = 5.3E-3 #Inercia pila 2, m4
249
250 # Parámetros de las barras:
251 EA_tablero = Elast * At
252 EI_tablero = Elast * It
253
254 EA_p1 = Elast * Ap1
255 EI_p1 = Elast * Ip1
256
257 EA_p2 = Elast * Ap2
258 EI_p2 = Elast * Ip2
259
260 #-----
261 # Definición de las barras
262 #-----
263 barra_a = P2D(p1,p2,EA_tablero,EI_tablero)
264 barra_b = P2D(p2,p3,EA_tablero,EI_tablero)
265 barra_c = P2D(p3,p4,EA_tablero,EI_tablero)
266 barra_d = P2D(p5,p2,EA_p1,EI_p1)
267 barra_e = P2D(p6,p3,EA_p2,EI_p2)
268
269 #-----
270 # Dibujo aproximado de la estructura
271 #-----
272 viga_x = [p1.x, p2.x, p3.x, p4.x]
273 viga_y = [p1.y, p2.y, p3.y, p4.y]
274 line, = plt.plot(viga_x, viga_y, label = "Viga", linewidth =
275                  2.0)
276 pilar1_x = [p5.x, p2.x]
276 pilar1_y = [p5.y, p2.y]
277 line, = plt.plot(pilar1_x, pilar1_y, label = "Pilar1", linewidth
278                  = 2.0)
278 pilar2_x = [p6.x, p3.x]

```



```

279 pilar2_y = [p6.y, p3.y]
280 line, = plt.plot(pilar2_x, pilar2_y, label = "Pilar2", linewidth
    = 2.0)
281
282 plt.axes().set_aspect('equal','datalim')
283 plt.show()
284
285 #-----
286 # Definición de las submatrices de rigidez necesarias
287 #-----
288 # Barra a: Condensación del giro en 1
289 #   Observe la definición de la función K_Condensa_Giro.
290 #   El índice 1 hace referencia a que se condensa el giro del
    nodo inicial. Si hubiese sido 2 se condensa el del extremo.
    Por defecto condensa en 2, por eso en este caso hay que
    indicarlo.
291
292 ka = barra_a.K_Condensa_Giro(1)
293 np.set_printoptions(precision=3)
294 print("")
295 print("Matriz de rigidez de la barra a, giro en 1 condensado")
296 print(ka)
297
298 # Extracción de la caja k22_a: Atención a los índices en Python
299 k22_a = ka[2:5,2:5]
300 # También se podía haber hecho directamente:
301 # k22_a = barra_a.K_Condensa_Giro(1)[2:5,2:5]
302
303
304 # Barra b: No se condensa nada
305 # Con la función específica de la clase se hace así:
306 k22_b_clase = barra_b.K_global_caja(1,1)
307 k23_b_clase = barra_b.K_global_caja(1,2)
308 k33_b_clase = barra_b.K_global_caja(2,2)
309
310 # Se han designado "clase" para indicar que se han obtenido con
    la función K_global_caja()
311 # Sin la función de la clase, se hace así:
312 k22_b = barra_b.K_global()[0:3,0:3]
313 k23_b = barra_b.K_global()[0:3,3:6]
314 k33_b = barra_b.K_global()[3:6,3:6]
315
316 # Observación: La operación de selección es inclusiva en el
    primer índice y exclusiva en el segundo 0:3 significa que
    tome los valores 0,1,2, y no el 3
317 # Si se comparan K22_b con K22_b_clase y siguientes, se observa
    que son iguales.
318
319
320 # Barra c: Condensación del giro en el nodo final
321 kc = barra_c.K_Condensa_Giro(2) # No es necesario indicar (2)
    pues está por defecto
322 k33_c = kc[0:3,0:3]
323 k34_c = kc[0:3,3:5] #Caja de dimensiones 3x2
324 k44_c = kc[3:5,3:5] #Caja de dimensiones 2x2
325

```

```

326
327 # Barra d: Condensación del giro en el nodo final
328
329 kd = barra_d.K_Condensa_Giro() # Observe que por defecto toma el
    giro condensado en 2
330
331 # Se expande a una 3x3
332 k22_d = np.zeros([3,3])
333 k22_d[0:2,0:2] = kd[3:5,3:5]
334
335
336 # Barra e: Condensación del giro en el extremo
337 ke = barra_e.K_Condensa_Giro() # Observe que por defecto toma el
    giro condensado en 2
338 # Se expande a una 3x3
339 k33_e = np.zeros([3,3])
340 k33_e[0:2,0:2] = ke[3:5,3:5]
341
342
343 # Suma de matrices en nudos 2 y 3
344 k22 = k22_a + k22_b + k22_d
345 k33 = k33_b + k33_c + k33_e
346
347 #-----
348 # Ensamblaje de la matriz de rigidez
349 #-----
350
351 krigidez = np.zeros([8,8])
352
353 krigidez[0:3,0:3] = k22
354
355 krigidez[0:3,3:6] = k23_b
356 krigidez[3:6,0:3] = k23_b.T # Completamos con la traspuesta
357
358 krigidez[3:6,3:6] = k33
359
360 krigidez[3:6,6:8] = k34_c
361 krigidez[6:8,3:6] = k34_c.T #Completamos con la traspuesta
362
363 krigidez[6:8,6:8] = k44_c
364
365 #-----
366 # Vector de fuerzas de empotramiento
367 #-----
368 q = 100 # Carga distribuida: 10 kN/m
369
370 La = longitud(p1,p2)
371 F2_a = np.array([0, 5./8 *q *La, -q*La**2./8.])
372
373 Lb = longitud(p2,p3)
374 F2_b = np.array([0, q * Lb / 2., q * Lb**2 / 12.])
375 F3_b = np.array([0, q * Lb / 2., -q * Lb**2 / 12.])
376
377 Lc = longitud(p3,p4)
378 F3_c = np.array([0, 5./8 *q*Lc, q * Lc**2/8.])
379 F4_c = np.array([0, 3./8.*q*Lc])

```

```

380
381 F2 = F2_a + F2_b
382 F3 = F3_b + F3_c
383
384 Fempotram = np.zeros(8)
385 Fempotram[0:3] = F2
386 Fempotram[3:6] = F3
387 Fempotram[6:8] = F4_c
388
389 # -----
390 # Eliminación de filas y columnas para imponer u4y = 0
391 # Como Python indexa los vectores desde 0, este gdl es el 7
392 # -----
393
394 krigidez_final = np.delete(krigidez,7,axis = 0)
395 krigidez_final = np.delete(krigidez_final,7,axis = 1)
396 Fuerzas = -1 * Fempotram # Cambio de signo porque el sistema es
    K u = F - Femp
397 Fuerzas = np.delete(Fuerzas,7,axis=0)
398
399 # -----
400 ## Resolución del sistema de ecuaciones
401 # -----
402 u = np.linalg.solve(krigidez_final,Fuerzas)
403 print("")
404 print("Solución del sistema ")
405 print("u_2x =", u[0], "m")
406 print("u_2y =", u[1], "m")
407 print("Theta_2 =",u[2], "rad")
408 print("")
409 print("u_3x =", u[3], "m")
410 print("u_3y =", u[4], "m")
411 print("Theta_3 =", u[5], "rad")
412 print("")
413 print("u_4x =", u[6], "m")

```

Los resultados que se obtienen son:

$$\begin{aligned}
 u_{2x} &= 1.509 \cdot 10^{-8} \text{ m}; & u_{2y} &= -1.331 \cdot 10^{-2} \text{ m}; & \theta_2^{ab} &= -9.501 \cdot 10^{-3} \text{ rad} \\
 u_{3x} &= -3.904 \cdot 10^{-4} \text{ m}; & u_{3y} &= -1.345 \cdot 10^{-2} \text{ m}; & \theta_3^{bc} &= -9.580 \cdot 10^{-3} \text{ rad} \\
 u_{4x} &= -3.905 \cdot 10^{-4} \text{ m}
 \end{aligned}$$

5.3. Apoyos no concordantes

Los apoyos no concordantes son aquellos que están orientados en direcciones que no coinciden con las de los ejes globales. Veamos el ejemplo del siguiente pórtico:

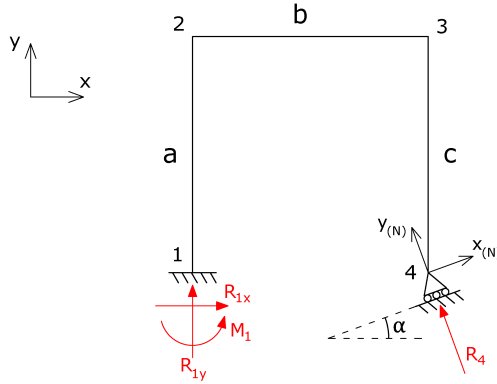


Figura 5.14: Estructura con apoyo no concordante.

Se observa que el apoyo del nudo 4 tiene una orientación no concordante con el sistema global, por lo que se ha definido un sistema de coordenadas nodales propio para dicho nudo, orientado en la dirección del apoyo.

Comencemos planteando el sistema de ecuaciones de la estructura del mismo modo que hemos hecho hasta ahora (se incluyen fuerzas nodales y de empotramiento genéricas, para mayor generalidad del ejemplo):

$$\begin{pmatrix} R_{1x} \\ R_{1y} \\ M_1 \\ F_{2x} \\ F_{2y} \\ M_2 \\ F_{3x} \\ F_{3y} \\ M_3 \\ R_{4x} \\ R_{4y} \\ 0 \end{pmatrix} - \begin{pmatrix} \tilde{\mathbf{P}}_1 \\ - \\ \tilde{\mathbf{P}}_2 \\ - \\ \tilde{\mathbf{P}}_3 \\ - \\ \tilde{\mathbf{P}}_4 \end{pmatrix} = \begin{pmatrix} k_{11}^a & k_{12}^a & 0 & 0 \\ k_{21}^a & k_{22}^{a+b} & k_{23}^b & 0 \\ 0 & k_{32}^b & k_{33}^{b+c} & k_{34}^c \\ 0 & 0 & k_{43}^c & k_{44}^c \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ u_{2x} \\ u_{2y} \\ \theta_2 \\ u_{3x} \\ u_{3y} \\ \theta_3 \\ u_{4x} \\ u_{4y} \\ \theta_4 \end{pmatrix} \quad (5.5)$$

Como se vio en el Capítulo 2, el cambio de coordenadas para vectores y matrices de nodales a globales es el siguiente:

$$\mathbf{F}_i = \mathbf{m}_{NG} \mathbf{F}_{i,N} \quad (5.6)$$

$$\mathbf{u}_i = \mathbf{m}_{NG} \mathbf{u}_{i,N} \quad (5.7)$$

siendo la matriz de cambio de coordenadas nodales a globales:

$$\mathbf{m}_{NG} = \begin{pmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

donde $c = \cos \alpha$, $s = \sin \alpha$, y α es el ángulo entre los ejes globales y nodales (inclinación del apoyo no concordante). La matriz de cambio de globales a nodales, por su parte, será la transpuesta de la matriz \mathbf{m}_{NG} .

Sabiendo esto, extraemos las filas del apoyo no concordante (filas 10, 11 y 12) del sistema de la Ecuación 5.5:

$$\mathbf{R}_4 - \tilde{\mathbf{P}}_4 = \mathbf{k}_{43}^c \mathbf{u}_3 + \mathbf{k}_{44}^c \mathbf{u}_4$$

y se transforma el desplazamiento a coordenadas nodales según la Ecuación 5.7:

$$\mathbf{R}_4 - \tilde{\mathbf{P}}_4 = \mathbf{k}_{43}^c \mathbf{u}_3 + \mathbf{k}_{44}^c \mathbf{m}_{NG} \mathbf{u}_{4,N} \quad (5.8)$$

Operando igualmente en cualquier otra fila, obtenemos el mismo resultado. Si vemos por ejemplo las filas correspondientes al nudo 3 (filas 7 a 9):

$$\mathbf{F}_3 - \tilde{\mathbf{P}}_3 = \mathbf{k}_{32}^b \mathbf{u}_2 + \mathbf{k}_{33}^{b+c} \mathbf{u}_3 + \mathbf{k}_{34}^c \mathbf{m}_{NG} \mathbf{u}_{4,N}$$

Es decir, podemos escribir el sistema como:

$$\begin{pmatrix} \mathbf{R}_1 \\ \mathbf{F}_2 \\ \mathbf{F}_3 \\ \mathbf{R}_4 \end{pmatrix} - \begin{pmatrix} \tilde{\mathbf{P}}_1 \\ \tilde{\mathbf{P}}_2 \\ \tilde{\mathbf{P}}_3 \\ \tilde{\mathbf{P}}_4 \end{pmatrix} = \begin{pmatrix} \mathbf{k}_{11}^a & \mathbf{k}_{12}^a & \mathbf{0} & \mathbf{0} \\ \mathbf{k}_{21}^a & \mathbf{k}_{22}^{a+b} & \mathbf{k}_{23}^b & \mathbf{0} \\ \mathbf{0} & \mathbf{k}_{32}^b & \mathbf{k}_{33}^{b+c} & \mathbf{k}_{34}^c \mathbf{m}_{NG} \\ \mathbf{0} & \mathbf{0} & \mathbf{k}_{43}^c & \mathbf{k}_{44}^c \mathbf{m}_{NG} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_{4,N} \end{pmatrix}$$

Ya tenemos el sistema con los desplazamientos en el apoyo no concordante en coordenadas nodales, pero nos interesa tener también así las reacciones y fuerzas de empotramiento en el mismo. Si multiplicamos la Ecuación 5.8 por \mathbf{m}_{GN} por la izquierda:

$$\begin{aligned} \mathbf{m}_{GN} [\mathbf{R}_4 - \tilde{\mathbf{P}}_4] &= \mathbf{m}_{GN} [\mathbf{k}_{43}^c \mathbf{u}_3 + \mathbf{k}_{44}^c \mathbf{m}_{NG} \mathbf{u}_{4,N}] \Rightarrow \\ \Rightarrow \mathbf{m}_{GN} \mathbf{R}_4 - \mathbf{m}_{GN} \tilde{\mathbf{P}}_4 &= \mathbf{m}_{GN} \mathbf{k}_{43}^c \mathbf{u}_3 + \mathbf{m}_{GN} \mathbf{k}_{44}^c \mathbf{m}_{NG} \mathbf{u}_{4,N} \end{aligned}$$

que, si tenemos en cuenta la Ecuación 5.6, es igual a:

$$\mathbf{R}_{4,N} - \tilde{\mathbf{P}}_{4,N} = \mathbf{m}_{GN} \mathbf{k}_{43}^c \mathbf{u}_3 + \mathbf{m}_{GN} \mathbf{k}_{44}^c \mathbf{m}_{NG} \mathbf{u}_{4,N}$$

Sustituyéndolo en el sistema de ecuaciones:

$$\begin{pmatrix} \mathbf{R}_1 \\ \mathbf{F}_2 \\ \mathbf{F}_3 \\ \mathbf{R}_{4,N} \end{pmatrix} - \begin{pmatrix} \tilde{\mathbf{P}}_1 \\ \tilde{\mathbf{P}}_2 \\ \tilde{\mathbf{P}}_3 \\ \tilde{\mathbf{P}}_{4,N} \end{pmatrix} = \begin{pmatrix} \mathbf{k}_{11}^a & \mathbf{k}_{12}^a & \mathbf{0} & \mathbf{0} \\ \mathbf{k}_{21}^a & \mathbf{k}_{22}^{a+b} & \mathbf{k}_{23}^b & \mathbf{0} \\ \mathbf{0} & \mathbf{k}_{32}^b & \mathbf{k}_{33}^{b+c} & \mathbf{k}_{34}^c \mathbf{m}_{NG} \\ \mathbf{0} & \mathbf{0} & \mathbf{m}_{GN} \mathbf{k}_{43}^c & \mathbf{m}_{GN} \mathbf{k}_{44}^c \mathbf{m}_{NG} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_{4,N} \end{pmatrix}$$

Con este sistema podremos resolver la estructura utilizando las fuerzas y desplazamientos en coordenadas nodales en el apoyo no concordante.

A modo de resumen, podemos decir que si hay un apoyo no concordante en el nudo i de la estructura, se deberá multiplicar por \mathbf{m}_{GN} por la izquierda la fila correspondiente al nudo i , y por \mathbf{m}_{NG} por la derecha la columna de \mathbf{K} correspondiente a ese mismo nudo.

5.4. Apoyos elásticos

Hasta este momento hemos considerado todos los apoyos como infinitamente rígidos, es decir, o permitían totalmente el desplazamiento en un cierto **gdl** o lo impedían por completo, pero esto no siempre es así. Por ejemplo, en el caso de estructuras muy rígidas cimentadas sobre suelos que puedan ser considerados elásticos o bien con una unión elástica estructura-cimentación, las condiciones de apoyo con el terreno deberán ser tratadas como elásticas.

La existencia de apoyos elásticos hace que la condición de contorno en el mismo venga expresada a modo de relación entre fuerzas y esfuerzos, lo cual se puede idealizar mediante muelles de una determinada **rigidez** k .

Tomemos un elemento genérico de una estructura que tenga un apoyo elástico en sus tres **gdl** en el nudo i :

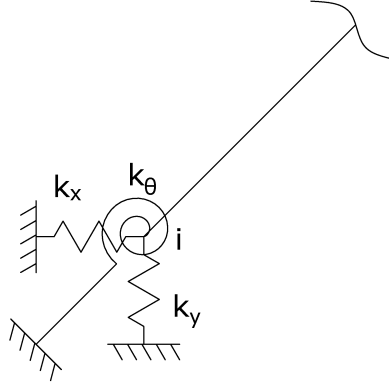


Figura 5.15: Apoyo elástico en sus 3 gdl.

Aplicando la Ley de Hooke, las fuerzas en el apoyo elástico serán:

$$\begin{aligned} F_{ix} &= -k_x u_{ix} \\ F_{iy} &= -k_y u_{iy} \\ M_i &= -k_\theta \theta_i \end{aligned}$$

Expresado de forma matricial:

$$\mathbf{F}^{\text{muelle}} = \begin{pmatrix} F_{ix} \\ F_{iy} \\ M_i \end{pmatrix} = - \begin{pmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_\theta \end{pmatrix} \begin{pmatrix} u_{ix} \\ u_{iy} \\ \theta_i \end{pmatrix} = -\mathbf{k}^{\text{muelle}} \mathbf{u}_i \quad (5.9)$$

Consideramos ahora el sistema global de la estructura sumándole la fuerza del muelle. Si vemos la fila correspondiente al nudo i :

$$\mathbf{F}_i + \mathbf{F}^{\text{muelle}} - \tilde{\mathbf{P}}_i = \sum_j (\mathbf{K}_{ij} \mathbf{u}_j)$$

Sustituyendo el valor de la fuerza del muelle por su expresión obtenida en la Ecuación 5.9:

$$\begin{aligned} \mathbf{F}_i - \mathbf{k}^{\text{muelle}} \mathbf{u}_i - \tilde{\mathbf{P}}_i &= \sum_j (\mathbf{K}_{ij} \mathbf{u}_j) \Rightarrow \\ \Rightarrow \mathbf{F}_i - \tilde{\mathbf{P}}_i &= \mathbf{k}^{\text{muelle}} \mathbf{u}_i + \sum_j (\mathbf{K}_{ij} \mathbf{u}_j) \Rightarrow \\ \Rightarrow \mathbf{F}_i - \tilde{\mathbf{P}}_i &= (\mathbf{k}^{\text{muelle}} + \mathbf{K}_{ii}) \mathbf{u}_i + \sum_{j \neq i} (\mathbf{K}_{ij} \mathbf{u}_j) \end{aligned}$$

Es decir, si hay un apoyo elástico en un nudo i de una estructura, la matriz de rigidez del muelle se deberá sumar al término \mathbf{K}_{ii} de la matriz de rigidez \mathbf{K} de la estructura, como se muestra en el siguiente ejemplo.

Ejercicio 5.3 Definir el sistema de ecuaciones de la siguiente estructura (forma simbólica), sabiendo que tiene dos muelles de rigideces $k_y = 2 \cdot 10^4 \text{ kN/m}$ y $k_\theta = 10^4 \text{ kN} \cdot \text{m/rad}$.

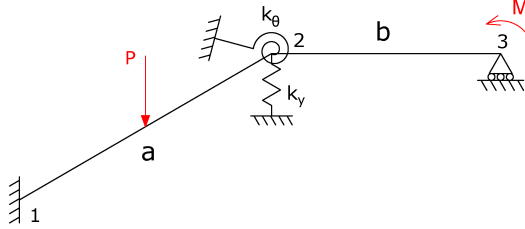


Figura 5.16: Ejercicio 5.3. Estructura con apoyo elástico.

Solución

Planteamos el sistema de ecuaciones de la estructura ($\mathbf{F} - \tilde{\mathbf{F}} = \mathbf{K}\mathbf{u}$). Según lo visto en el Apartado 5.2 no se cumplen las condiciones para condensar ningún gdl.

$$\begin{pmatrix} R_{1x} \\ R_{1y} \\ M_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ R_{3y} \\ M \end{pmatrix} - \begin{pmatrix} \tilde{\mathbf{P}}_1^a \\ \\ \tilde{\mathbf{P}}_2^a \\ \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{k}_{11}^a & \mathbf{k}_{12}^a & \mathbf{0} \\ \vdots & \vdots & \vdots \\ \mathbf{k}_{21}^a & \mathbf{k}_{22}^{a+b+m} & \mathbf{k}_{23}^b \\ \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{k}_{32}^b & \mathbf{k}_{33}^b \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ u_{2x} \\ u_{2y} \\ \theta_2 \\ u_{3x} \\ 0 \\ \theta_3 \end{pmatrix}$$

Donde

$$\mathbf{k}_{22}^{a+b+m} = \mathbf{k}_{22}^a + \mathbf{k}_{22}^b + \mathbf{k}^{muelle}$$

Siendo \mathbf{k}^{muelle} la matriz de rigidez del apoyo elástico en el nudo 2:

$$\mathbf{k}^{muelle} = \begin{pmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_\theta \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 \cdot 10^4 & 0 \\ 0 & 0 & 10^4 \end{pmatrix}$$

A partir de este punto, la resolución de la estructura mediante el MDR se realizaría de la misma forma que se ha expuesto en capítulos anteriores.

Apoyos elásticos no concordantes

En los casos en que los muelles no sigan las direcciones del sistema de coordenadas global, estos *no* se pueden descomponer directamente en sus “componentes” en el sistema global.

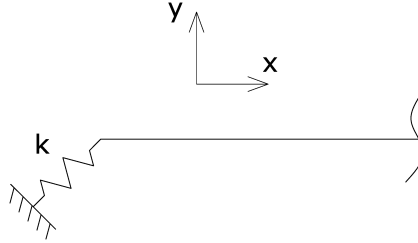


Figura 5.17: Apoyo elástico no concordante con el sistema de coordenadas global.

En estos casos hay dos opciones para resolver el problema:

1. Convertir la **matriz del muelle a coordenadas globales** y resolver el problema del modo habitual en coordenadas globales.

$$\mathbf{k}^{muelle} = \mathbf{m}_{NG} \mathbf{k}_N^{muelle} \mathbf{m}_{GN}$$

Este es, en general, el modo de proceder más sencillo.

2. Tratar el apoyo elástico como un **apoyo no concordante**, operando del modo descrito en el Apartado 5.3.

5.5. Ligaduras entre grados de libertad

Es normal encontrarse con estructuras que tienen ciertos **gdl**s relacionados entre sí. Esta situación ocurrirá cuando dos o más barras se encuentren conectadas por un elemento con la suficiente rigidez como para ser tratado como **sólido rígido**. Este sólido rígido hará que los desplazamientos de los nudos que conecten con él no sean independientes, sino que estén ligados entre sí.

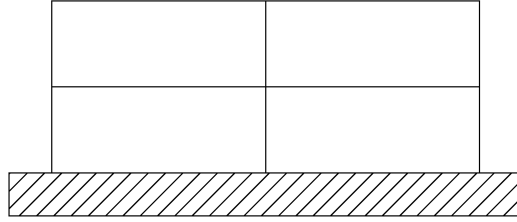


Figura 5.18: Ejemplo de ligadura entre **gd1**: pórtico con cimentación en losa.

La **ecuación de ligadura** entre **gd1s** la expresaremos de la forma:

$$\mathbf{u} = \mathbf{T}\mathbf{u}_i \quad (5.10)$$

Donde \mathbf{u} es el vector de los desplazamientos que intervienen en el problema, \mathbf{u}_i es el vector de desplazamientos independientes y \mathbf{T} es la **matriz de ligadura**, que relaciona los desplazamientos con los desplazamientos independientes. Esta matriz \mathbf{T} se podrá obtener mediante ecuaciones de compatibilidad de desplazamientos en el sólido rígido.

Si partimos del sistema de ecuaciones de una estructura y aplicamos la expresión de la Ecuación 5.10, tenemos:

$$\mathbf{F} - \tilde{\mathbf{F}} = \mathbf{K}\mathbf{u} \Rightarrow \mathbf{F} - \tilde{\mathbf{F}} = \mathbf{K}\mathbf{T}\mathbf{u}_i$$

Para poder seguir trabajando con matrices simétricas, como hasta ahora, basta con multiplicar la ecuación anterior por la traspuesta de \mathbf{T} por la izquierda:

$$\mathbf{T}^T\mathbf{F} - \mathbf{T}^T\tilde{\mathbf{F}} = \mathbf{T}^T\mathbf{K}\mathbf{T}\mathbf{u}_i \quad (5.11)$$

Aplicando las ecuaciones de equilibrio en el sólido rígido en la expresión $\mathbf{T}^T\mathbf{F}$ y resolviendo el sistema de ecuaciones 5.11 se podrán calcular los desplazamientos de la estructura que son independientes entre sí.

De forma resumida, el procedimiento para resolver una estructura con ligaduras entre **gd1** es el siguiente:

1. **Aislar el sólido rígido (SR)** de la estructura de barras. En los nudos de las barras que conectan con el **SR** aparecerán unas reacciones iguales y de signo contrario a las que aparecerán en puntos correspondientes del **SR**.
2. Plantear el sistema de ecuaciones de la estructura, $\mathbf{F} - \tilde{\mathbf{F}} = \mathbf{K}\mathbf{u}$. En el vector de fuerzas externas \mathbf{F} aparecerán las reacciones establecidas en el punto anterior.
3. Plantear las **ecuaciones de compatibilidad** de desplazamientos en el **SR**, determinando cuáles de estos desplazamientos son independientes entre sí (\mathbf{u}_i).

4. Plantear las **ecuaciones de equilibrio** en el SR.
5. A partir de las ecuaciones de compatibilidad del SR, determinar la matriz de ligadura \mathbf{T} tal que $\mathbf{u} = \mathbf{T}\mathbf{u}_i$
6. Calcular el producto $\mathbf{T}^T \mathbf{F}$ y aplicar al resultado las ecuaciones de equilibrio del SR.
7. Calcular las fuerzas de empotramiento y a continuación el producto $\mathbf{T}^T \tilde{\mathbf{F}}$
8. Calcular los términos de la matriz de rigidez que intervienen en el sistema y operar para obtener la expresión de $\mathbf{T}^T \mathbf{K} \mathbf{T}$
9. Plantear el sistema de la Ecuación 5.11 y resolverlo para obtener los **desplazamientos independientes**.
10. Calcular los esfuerzos y reacciones de la forma habitual.

A continuación resolveremos un ejemplo de estructura con ligadura entre gdl.

Ejercicio 5.4 *Calcular, utilizando el Método Directo de la Rigidez, el giro del sólido rígido de la siguiente estructura:*

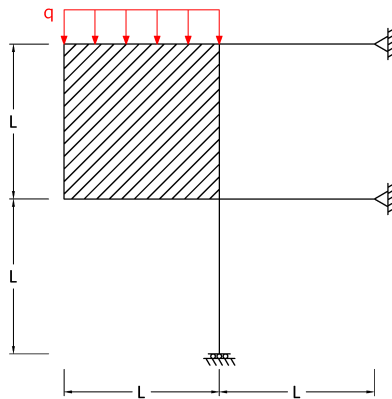


Figura 5.19: Ejercicio 5.1. Estructura con sólido rígido.

Datos: $EA = 1.2 \cdot 10^5 \text{ kN}$; $EI = 200 \text{ kN} \cdot \text{m}^2$; $q = 15 \text{ kN/m}$; $L = 5 \text{ m}$.

Solución

En primer lugar, aislamos el sólido rígido (SR) de la estructura de barras, apareciendo una serie de reacciones en los nudos de unión. Numeramos, además, los nudos y las barras.

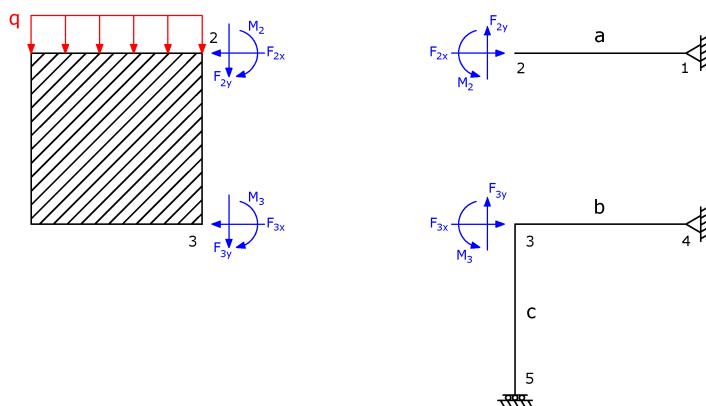


Figura 5.20: Ejercicio 5.1. Separación SR – estructura de barras.

Establecemos los **gdl** en los sistemas global y local para la estructura de barras. Condensaremos los **gdl** que cumplen las condiciones indicadas en el Apartado 5.2, es decir, giros y desplazamientos perpendiculares a las barras que estén liberados y que no tengan cargas nodales externas aplicadas. Los **gdl** que cumplen estas condiciones con θ_1 , θ_4 , y u_{5x} .

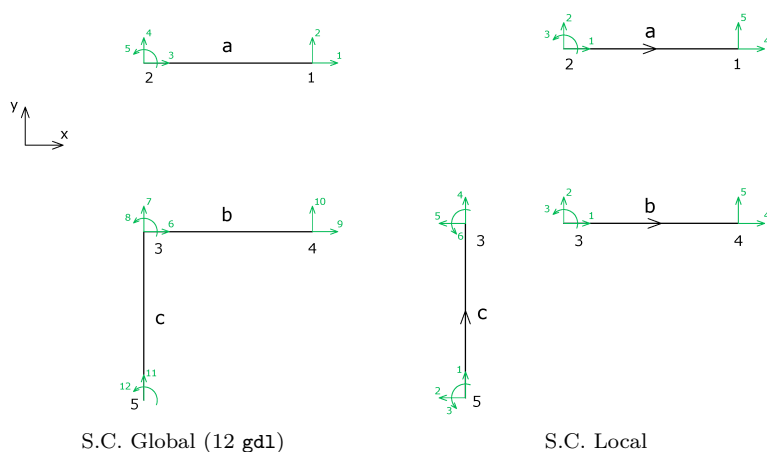


Figura 5.21: Ejercicio 5.1. **gdl** en la estructura de barras (cond. θ_1 , θ_4 , y u_{5x}).

Planteamos el sistema de ecuaciones de la estructura de barras (condensados θ_1 , θ_4 , y u_{5x}). Al no haber fuerzas de empotramiento en este caso,

tenemos que $\mathbf{F} = \mathbf{K}\mathbf{u}$:

$$\begin{pmatrix} R_{1x} \\ R_{1y} \\ F_{2x} \\ F_{2y} \\ M_2 \\ F_{3x} \\ F_{3y} \\ M_3 \\ R_{4x} \\ R_{4y} \\ R_{5y} \\ M_5 \end{pmatrix} = \begin{pmatrix} \mathbf{k}_{11}^a & \mathbf{k}_{12}^a & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{k}_{21}^a & \mathbf{k}_{22}^a & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{k}_{33}^{b+c} & \mathbf{k}_{34}^b & \mathbf{k}_{35}^c \\ \mathbf{0} & \mathbf{0} & \mathbf{k}_{43}^b & \mathbf{k}_{44}^b & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{k}_{53}^c & \mathbf{0} & \mathbf{k}_{55}^c \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ u_{2x} \\ u_{2y} \\ \theta_2 \\ u_{3x} \\ u_{3y} \\ \theta_3 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Podemos observar que el sistema tiene un total de 12 ecuaciones y 18 incógnitas. Las 6 ecuaciones que faltan se obtendrán de plantear el equilibrio y la compatibilidad en el SR.

Procediendo de la forma habitual, extraemos las ecuaciones del sistema que tienen los desplazamientos como incógnitas (ecuaciones 3 a 8). Este será el sistema que resolvamos:

$$\begin{pmatrix} \mathbf{F}_2 \\ \mathbf{F}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{k}_{22}^a & \mathbf{0} \\ \mathbf{0} & \mathbf{k}_{33}^{b+c} \end{pmatrix} \begin{pmatrix} \mathbf{u}_2 \\ \mathbf{u}_3 \end{pmatrix}$$

Obtenemos, como hemos dicho, las ecuaciones que nos faltan:

- Ecuaciones de compatibilidad de desplazamientos en el SR.

$$u_{3x} = u_{2x} + \theta_2 L$$

$$u_{3y} = u_{2y}$$

$$\theta_3 = \theta_2$$

- Ecuaciones de equilibrio en el SR.

$$\sum F_x = 0 \Rightarrow F_{2x} + F_{3x} = 0$$

$$\sum F_y = 0 \Rightarrow F_{2y} + F_{3y} = -qL$$

$$\sum M_{[2]} = 0 \Rightarrow M_2 + M_3 + F_{3x}L = \frac{qL^2}{2}$$

A partir de las ecuaciones de compatibilidad, podemos definir la matriz de ligadura \mathbf{T} . Se observa que los desplazamientos del nudo 2 están ligados con los del nudo 3; se opta por considerar expresar los desplazamientos en el nudo 3 en función de los del 2, por lo que el vector de desplazamientos independientes coincidirá con \mathbf{u}_2 .

$$\mathbf{u} = \mathbf{T}\mathbf{u}_i \Rightarrow \begin{pmatrix} u_{2x} \\ u_{2y} \\ \theta_2 \\ u_{3x} \\ u_{3y} \\ \theta_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & L \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_{2x} \\ u_{2y} \\ \theta_2 \end{pmatrix}$$

La matriz \mathbf{T} , de dimensiones 6×3 , puede descomponerse en dos matrices de 3×3 : una matriz \mathbf{T}_2 que liga los desplazamientos en 2 con los desplazamientos independientes (en este caso es igual a la matriz identidad) y una matriz \mathbf{T}_3 que liga los desplazamientos en 3 con los desplazamientos independientes.

$$\mathbf{T}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \mathbf{I}; \quad \mathbf{T}_3 = \begin{pmatrix} 1 & 0 & L \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

De este modo, podemos escribir la ecuación de ligadura como:

$$\mathbf{u} = \mathbf{T}\mathbf{u}_i \Rightarrow \begin{pmatrix} \mathbf{u}_2 \\ \mathbf{u}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{I} \\ \mathbf{T}_3 \end{pmatrix} \mathbf{u}_2 = \begin{pmatrix} \mathbf{u}_2 \\ \mathbf{T}_3\mathbf{u}_2 \end{pmatrix}$$

Calculamos ahora los otros términos del sistema $\mathbf{T}^T \mathbf{F} = \mathbf{T}^T \mathbf{K} \mathbf{T} \mathbf{u}_i$. Comenzamos por el vector de fuerzas:

$$\mathbf{T}^T \mathbf{F} = \begin{pmatrix} \mathbf{I} & \mathbf{T}_3^T \end{pmatrix} \begin{pmatrix} \mathbf{F}_2 \\ \mathbf{F}_3 \end{pmatrix} = \mathbf{F}_2 + \mathbf{T}_3^T \mathbf{F}_3 = \begin{pmatrix} F_{2x} + F_{3x} \\ F_{2y} + F_{3y} \\ M_2 + M_3 + F_{3x}L \end{pmatrix}$$

Aplicando las ecuaciones de equilibrio, queda:

$$\mathbf{T}^T \mathbf{F} = \begin{pmatrix} 0 \\ -qL \\ \frac{qL^2}{2} \end{pmatrix} = \begin{pmatrix} 0 \\ -75 \\ 187.5 \end{pmatrix}$$

Por lo que respecta a la matriz de rigidez, tenemos:

$$\begin{aligned} \mathbf{T}^T \mathbf{K} \mathbf{T} &= \left(\mathbf{I} \mid \mathbf{T}_3^T \right) \left(\begin{array}{c|c} \mathbf{k}_{22}^a & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{k}_{33}^{b+c} \end{array} \right) \left(\begin{array}{c} \mathbf{I} \\ \hline \mathbf{T}_3 \end{array} \right) = \\ &= \left(\mathbf{I} \mid \mathbf{T}_3^T \right) \left(\begin{array}{c} \mathbf{k}_{22}^a \\ \hline \mathbf{k}_{33}^{b+c} \mathbf{T}_3 \end{array} \right) = \mathbf{k}_{22}^a + \mathbf{T}_3^T \mathbf{k}_{33}^{b+c} \mathbf{T}_3 \end{aligned}$$

Donde los términos de \mathbf{K} necesarios son los siguientes:

- Barras a y b ($L = 5 \text{ m}$; $\alpha = 0^\circ$). Condensado θ_j ¹.

$$\mathbf{k}_{22}^a = \mathbf{k}_{33}^b = \begin{pmatrix} \frac{EA}{L} & 0 & 0 \\ 0 & \frac{3EI}{L^3} & \frac{3EI}{L^2} \\ 0 & \frac{3EI}{L^2} & \frac{3EI}{L} \end{pmatrix} = \begin{pmatrix} 2.4 \cdot 10^4 & 0 & 0 \\ 0 & 4.8 & 24 \\ 0 & 24 & 120 \end{pmatrix}$$

- Barra c ($L = 5 \text{ m}$; $\alpha = 90^\circ$). Condensado u_{ix} . Si llamamos $c = \cos \alpha$ y $s = \sin \alpha$:

$$\mathbf{k}_{33}^c = \begin{pmatrix} c^2 \frac{EA}{L} & cs \frac{EA}{L} & 0 \\ cs \frac{EA}{L} & c^2 \frac{EA}{L} & 0 \\ 0 & 0 & \frac{EI}{L} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2.4 \cdot 10^4 & 0 \\ 0 & 0 & 40 \end{pmatrix}$$

Sustituyendo estos valores en la expresión anterior:

$$\mathbf{T}^T \mathbf{K} \mathbf{T} = \mathbf{k}_{22}^a + \mathbf{T}_3^T \mathbf{k}_{33}^{b+c} \mathbf{T}_3 = \begin{pmatrix} 4.8 \cdot 10^4 & 0 & 1.2 \cdot 10^5 \\ 0 & 2.4 \cdot 10^4 & 48 \\ 1.2 \cdot 10^5 & 48 & 6 \cdot 10^5 \end{pmatrix}$$

Finalmente, montamos el sistema y resolvemos:

$$\mathbf{T}^T \mathbf{F} = \mathbf{T}^T \mathbf{K} \mathbf{T} \mathbf{u}_i \Rightarrow \begin{pmatrix} 0 \\ -75 \\ 187.5 \end{pmatrix} = \begin{pmatrix} 4.8 \cdot 10^4 & 0 & 1.2 \cdot 10^5 \\ 0 & 2.4 \cdot 10^4 & 48 \\ 1.2 \cdot 10^5 & 48 & 6 \cdot 10^5 \end{pmatrix} \begin{pmatrix} u_{2x} \\ u_{2y} \\ \theta_2 \end{pmatrix}$$

¹La expresión de la matriz de rigidez de un elemento P2D con el giro en el nudo final condensado se puede obtener aplicando las ecuaciones vistas en el Apartado 5.2; siendo también posible encontrarla resuelta en diversos prontuarios de Cálculo Matricial, entre ellos el incluido en el Anexo B.

Se obtiene:

$$\begin{aligned} u_{2x} &= -1.56 \cdot 10^{-3} \text{ m} \\ u_{2y} &= -3.13 \cdot 10^{-3} \text{ m} \\ \theta_2 &= 6.25 \cdot 10^{-4} \text{ rad} \end{aligned}$$

Por compatibilidad de desplazamientos, el giro del sólido rígido que nos piden es igual al giro de los nudos 2 y 3:

$$\theta_{SR} = \theta_2 \Rightarrow \theta_{SR} = 6.25 \cdot 10^{-4} \text{ rad}$$

Si quisiéramos obtener los desplazamientos en el nudo 3, bastaría aplicar las ecuaciones de compatibilidad de desplazamientos en el **SR** o, lo que es lo mismo, la ecuación de ligadura. En este caso:

$$\mathbf{u}_3 = \mathbf{T}_3 \mathbf{u}_i = \mathbf{T}_3 \mathbf{u}_2 = \begin{pmatrix} 1 & 0 & L \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1.56 \cdot 10^{-3} \\ -3.13 \cdot 10^{-3} \\ 6.25 \cdot 10^{-4} \end{pmatrix} = \begin{pmatrix} 1.56 \cdot 10^{-3} \\ -3.13 \cdot 10^{-3} \\ 6.25 \cdot 10^{-4} \end{pmatrix}$$

5.6. Subestructuras

En estructuras de gran complejidad, con numerosos **gdl**s, es común que un determinado módulo con unas características concretas se repita múltiples veces, o que una serie de módulos distintos se conecten de diferentes formas para componer la estructura en su conjunto.

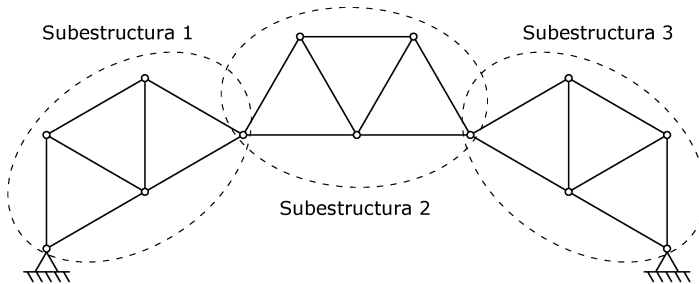


Figura 5.22: Estructura compuesta por subestructuras.

En estos casos, es útil definir cada módulo como una subestructura independiente, calculando su matriz de rigidez y condensando todos los **gdl** excepto los que conectan las subestructuras entre sí. Es decir, estamos tratando cada subestructura como si fuera un **macroelemento** de la estructura global.

El procedimiento de cálculo es el siguiente:

1. Calcular las matrices de rigidez de cada subestructura.
2. Condensar, para cada subestructura, todos los **gdl** que no estén en contacto con otra subestructura.
3. Transformar al mismo sistema de coordenadas todas las matrices calculadas.
4. Montar el sistema de ecuaciones de la estructura. Cada subestructura hará la función de un elemento.
5. Resolver el sistema para obtener los desplazamientos en los nudos que conectan subestructuras.
6. Calcular, si es necesario, los desplazamientos condensados.
7. Calcular los esfuerzos y reacciones.

5.7. Elementos especiales

En este texto se han tratado hasta el momento dos tipos de elementos, fundamentalmente: barras rectas de pórtico y tirantes o elementos articulados. Existen otros tipos de elementos cuyas matrices de rigidez puede ser útil conocer para resolver determinados problemas estructurales aplicando el Cálculo Matricial del modo descrito. A continuación se deducen las matrices de rigidez de algunos de estos elementos que denominaremos “elementos especiales”.

Barra con rótula en el centro

Consideramos el caso de un elemento recto en el plano, con 3 **gdl** por nudo, que contiene una rótula en su punto medio.

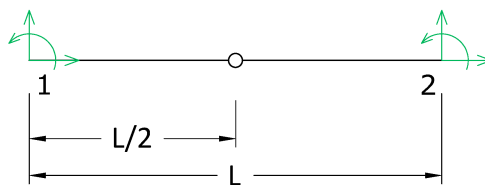


Figura 5.23: Barra con rótula en su punto medio, con 3 **gdl** en cada nudo.

En este caso se trata de un problema de condensación estática, donde partimos de dos barras tipo P2D unidas por una rótula en la que deberemos condensar los 4 **gdl**s de la misma (desplazamiento horizontal, desplazamiento vertical, giro por la izquierda y giro por la derecha). El resultado de la

condensación de los giros es conocido, ya que por la izquierda tenemos una barra a con el giro final condensado y por la derecha una barra b con el giro inicial condensado, cuyas matrices de rigidez se muestran en el Anexo B. Es decir, partimos de la siguiente estructura compuesta por dos barras:

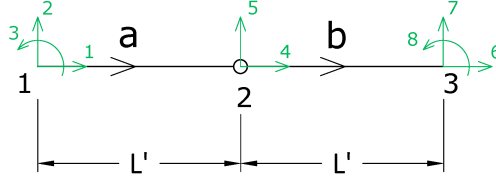


Figura 5.24: Elementos P2D unidos por una rótula, con los giros condensados en esa unión.

Por comodidad a la hora de operar, hemos llamado $L' = L/2$ a la longitud de cada barra. La matriz de rigidez de esta estructura (de dimensiones 8×8) será:

$$K = \begin{pmatrix} k_{11}^a & k_{12}^a & 0 \\ k_{21}^a & k_{22}^{a+b} & k_{23}^b \\ 0 & k_{32}^b & k_{33}^b \end{pmatrix} \quad (5.12)$$

donde las matrices elementales de las barras son las siguientes:

- Barra a (P2D condensado θ_j)

$$k^a = \begin{pmatrix} k_{11}^a & k_{12}^a \\ k_{21}^a & k_{22}^a \end{pmatrix} = \begin{pmatrix} \frac{EA}{L'} & 0 & 0 & -\frac{EA}{L'} & 0 \\ 0 & \frac{3EI}{L'^3} & \frac{3EI}{L'^2} & 0 & -\frac{3EI}{L'^3} \\ 0 & \frac{3EI}{L'^2} & \frac{3EI}{L'} & 0 & -\frac{3EI}{L'^2} \\ -\frac{EA}{L'} & 0 & 0 & \frac{EA}{L'} & 0 \\ 0 & -\frac{3EI}{L'^3} & -\frac{3EI}{L'^2} & 0 & \frac{3EI}{L'^3} \end{pmatrix}$$

- Barra b (P2D condensado θ_i)

$$k^b = \begin{pmatrix} k_{22}^b & k_{23}^b \\ k_{32}^b & k_{33}^b \end{pmatrix} = \begin{pmatrix} \frac{EA}{L'} & 0 & -\frac{EA}{L'} & 0 & 0 \\ 0 & \frac{3EI}{L'^3} & 0 & -\frac{3EI}{L'^3} & \frac{3EI}{L'^2} \\ -\frac{EA}{L'} & 0 & \frac{EA}{L'} & 0 & 0 \\ 0 & -\frac{3EI}{L'^3} & 0 & \frac{3EI}{L'^3} & -\frac{3EI}{L'^2} \\ 0 & \frac{3EI}{L'^2} & 0 & -\frac{3EI}{L'^2} & \frac{3EI}{L'} \end{pmatrix}$$

Sustituyendo los valores de las matrices elementales en la matriz de la estructura (ecuación 5.12):

$$\mathbf{K} = \begin{pmatrix} \frac{EA}{L'} & 0 & 0 & -\frac{EA}{L'} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{3EI}{L'^3} & \frac{3EI}{L'^2} & 0 & -\frac{3EI}{L'^3} & 0 & 0 & 0 & 0 \\ 0 & \frac{3EI}{L'^2} & \frac{3EI}{L'} & 0 & -\frac{3EI}{L'^2} & 0 & 0 & 0 & 0 \\ -\frac{EA}{L'} & 0 & 0 & \frac{2EA}{L'} & 0 & -\frac{EA}{L'} & 0 & 0 & 0 \\ 0 & -\frac{3EI}{L'^3} & -\frac{3EI}{L'^2} & 0 & \frac{6EI}{L'^3} & 0 & -\frac{3EI}{L'^3} & \frac{3EI}{L'^2} & 0 \\ 0 & 0 & 0 & -\frac{EA}{L'} & 0 & \frac{EA}{L'} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{3EI}{L'^3} & 0 & \frac{3EI}{L'^3} & -\frac{3EI}{L'^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{3EI}{L'^2} & 0 & -\frac{3EI}{L'^2} & \frac{3EI}{L'} & 0 \end{pmatrix}$$

De esta matriz queremos condensar los **gdl** 4 y 5, correspondientes a los desplazamientos en la rótula. Tal y como se demostró en el apartado 5.2, la expresión de la matriz de rigidez condensada es $\mathbf{K}^q = \mathbf{K}_{AA} - \mathbf{K}_{AB} \mathbf{K}_{BB}^{-1} \mathbf{K}_{BA}$ (ecuación 5.4), donde A representa a los **gdl** a mantener y B a los **gdl** a condensar.

Identifiquemos cada uno de los términos de esa expresión:

$$\mathbf{K}_{AA} = \begin{pmatrix} \frac{EA}{L'} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{3EI}{L'^3} & \frac{3EI}{L'^2} & 0 & 0 & 0 \\ 0 & \frac{3EI}{L'^2} & \frac{3EI}{L'} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{EA}{L'} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{3EI}{L'^3} & -\frac{3EI}{L'^2} \\ 0 & 0 & 0 & 0 & -\frac{3EI}{L'^2} & \frac{3EI}{L'} \end{pmatrix}$$

$$\mathbf{K}_{BB} = \begin{pmatrix} \frac{2EA}{L'} & 0 \\ 0 & \frac{6EI}{L'^3} \end{pmatrix}$$

$$\mathbf{K}_{AB} = \begin{pmatrix} -\frac{EA}{L'} & 0 \\ 0 & -\frac{3EI}{L'^3} \\ 0 & -\frac{3EI}{L'^2} \\ -\frac{EA}{L'} & 0 \\ 0 & -\frac{3EI}{L'^3} \\ 0 & \frac{3EI}{L'^2} \end{pmatrix}$$

$$\mathbf{K}_{BA} = \begin{pmatrix} -\frac{EA}{L'} & 0 & 0 & | & -\frac{EA}{L'} & 0 & 0 \\ 0 & -\frac{3EI}{L'^3} & -\frac{3EI}{L'^2} & | & 0 & -\frac{3EI}{L'^3} & \frac{3EI}{L'^2} \end{pmatrix}$$

Operamos para obtener la matriz condensada y finalmente sustituimos L' por $L/2$:

$$\mathbf{k}^q = \mathbf{K}_{AA} - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{BA} \Rightarrow$$

$$\Rightarrow \mathbf{k}^q = \begin{pmatrix} \frac{EA}{L} & 0 & 0 & | & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & | & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{3EI}{L} & | & 0 & -\frac{6EI}{L^2} & \frac{3EI}{L} \\ -\frac{EA}{L} & 0 & 0 & | & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & | & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{3EI}{L} & | & 0 & -\frac{6EI}{L^2} & \frac{3EI}{L} \end{pmatrix}$$

La expresión obtenida corresponde a la matriz de rigidez de una barra tipo P2D, de longitud L , con una rótula en su punto medio.

De forma análoga podríamos calcular la expresión general para una barra con una rótula en cualquier punto intermedio:

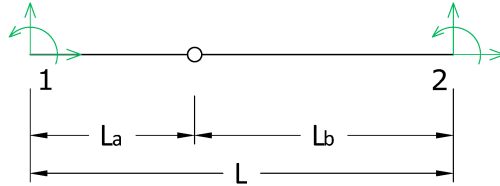


Figura 5.25: Barra con rótula en cualquier punto intermedio.

Obteniéndose la siguiente matriz de rigidez:

$$\mathbf{k} = \begin{pmatrix} \frac{EA}{L} & 0 & 0 & | & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{3EI}{L_a^3+L_b^3} & \frac{3EIL_a}{L_a^3+L_b^3} & | & 0 & -\frac{3EI}{L_a^3+L_b^3} & \frac{3EIL_b}{L_a^3+L_b^3} \\ 0 & \frac{3EIL_a}{L_a^3+L_b^3} & \frac{3EIL_a^2}{L_a^3+L_b^3} & | & 0 & -\frac{3EIL_a}{L_a^3+L_b^3} & \frac{3EIL_aL_b}{L_a^3+L_b^3} \\ -\frac{EA}{L} & 0 & 0 & | & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{3EI}{L_a^3+L_b^3} & -\frac{3EIL_a}{L_a^3+L_b^3} & | & 0 & \frac{3EI}{L_a^3+L_b^3} & -\frac{3EIL_b}{L_a^3+L_b^3} \\ 0 & \frac{3EIL_b}{L_a^3+L_b^3} & \frac{3EIL_aL_b}{L_a^3+L_b^3} & | & 0 & -\frac{3EIL_b}{L_a^3+L_b^3} & \frac{3EIL_b^2}{L_a^3+L_b^3} \end{pmatrix}$$

Barra con extremos flexibles

Tenemos ahora el caso de un elemento con extremos flexibles, es decir, el giro en ambos extremos está condicionado por sendos muelles de rigideces $k_{\theta 1}$ y $k_{\theta 2}$, como se muestra en la siguiente figura:

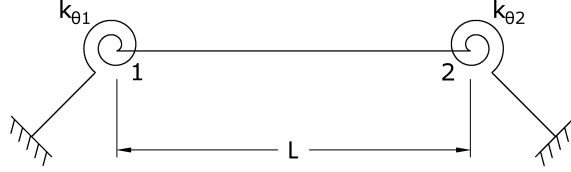


Figura 5.26: Elemento con extremos flexibles.

Definimos los siguientes términos:

$$\begin{aligned}\alpha &= 2 + \frac{6EI}{Lk_{\theta 1}}; & \beta &= 2 + \frac{6EI}{Lk_{\theta 2}}; & \gamma &= \frac{L^2 A}{I} \\ k_1 &= \frac{6\beta}{\alpha\beta - 1}; & k_2 &= \frac{6}{\alpha\beta - 1}; & k_3 &= \frac{6\alpha}{\alpha\beta - 1}; \\ k_4 &= k_1 + k_2 = \frac{6(\beta + 1)}{\alpha\beta - 1}; & k_5 &= k_2 + k_3 = \frac{6(\alpha + 1)}{\alpha\beta - 1}; \\ k_6 &= k_4 + k_5 = \frac{6(\alpha + \beta + 1)}{\alpha\beta - 1}\end{aligned}$$

Y tenemos que la matriz de rigidez de la barra queda:

$$\mathbf{k} = \frac{EI}{L^2} \begin{pmatrix} \gamma & 0 & 0 & -\gamma & 0 & 0 \\ 0 & k_6 & Lk_4 & 0 & -k_6 & Lk_5 \\ 0 & Lk_4 & L^2 k_1 & 0 & -Lk_4 & L^2 k_2 \\ -\gamma & 0 & 0 & \gamma & 0 & 0 \\ 0 & -k_6 & -Lk_4 & 0 & k_6 & -Lk_5 \\ 0 & Lk_5 & L^2 k_2 & 0 & -Lk_5 & L^2 k_3 \end{pmatrix} \quad (5.13)$$

En el caso particular en que las rigideces de ambos muelles sean iguales ($k_{\theta 1} = k_{\theta 2} = k_{\theta}$), los parámetros definidos anteriormente se simplifican:

$$\begin{aligned}\alpha &= \beta = 2 + \frac{6EI}{Lk_{\theta}} \\ k_1 &= k_3 = \frac{6\alpha}{\alpha^2 - 1}; & k_2 &= \frac{6}{\alpha^2 - 1}; \\ k_4 &= k_5 = k_1 + k_2 = \frac{6}{\alpha - 1}; & k_6 &= 2k_4 = \frac{12}{\alpha - 1}\end{aligned}$$

pudiéndose sustituir estos términos en la ecuación 5.13.

Barra con extremos rígidos

Consideramos en este caso una barra con los extremos rígidos, es decir, compuesta por dos tramos rígidos, uno en cada extremo, de longitudes αL y βL , y un segmento elástico entre ellos de longitud L .

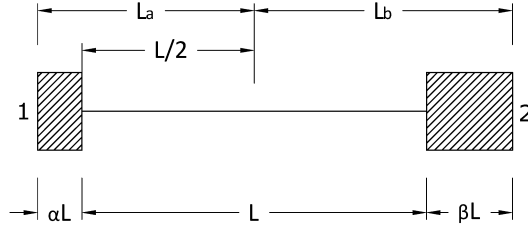


Figura 5.27: Elemento con extremos rígidos.

donde α y β son constantes y, por geometría:

$$L_a = L(\alpha + 1/2); \quad L_b = L(\beta + 1/2)$$

Además, definimos:

$$\gamma = \frac{L^2 A}{12I}; \quad \eta = \frac{L^2}{12}$$

La matriz de rigidez que se obtiene, aplicando las técnicas vistas en este capítulo es,

$$\mathbf{k} = \frac{12EI}{L^3} \left(\begin{array}{ccc|ccc} \gamma & 0 & 0 & -\gamma & 0 & 0 \\ 0 & 1 & L_a & 0 & -1 & L_b \\ 0 & L_a & L_a^2 + \eta & 0 & -L_a & L_a L_b - \eta \\ \hline -\gamma & 0 & 0 & \gamma & 0 & 0 \\ 0 & -1 & -L_a & 0 & 1 & -L_b \\ 0 & L_b & L_a L_b - \eta & 0 & -L_b & L_b^2 + \eta \end{array} \right)$$

En el caso particular en el que ambos extremos rígidos tengan la misma longitud, tendremos que $\alpha = \beta$ y por tanto $L_a = L_b$.

Anexo A

Matrices de rigidez elementales

En el presente Anexo se incluyen las matrices de rigidez elementales en coordenadas locales y globales para los elementos de las tipologías de estructuras de barras bidimensionales más comunes: estructura articulada plana, pórtico plano y emparrillado plano.

A.1. Estructura articulada plana (A2D)

Coordenadas locales

$$\mathbf{k}_L = \left(\begin{array}{c|c} \frac{EA}{L} & -\frac{EA}{L} \\ \hline -\frac{EA}{L} & \frac{EA}{L} \end{array} \right)$$

Coordenadas globales

$$\mathbf{k} = \frac{EA}{L} \left(\begin{array}{cc|cc} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ \hline -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{array} \right)$$

A.2. Pórtico plano (P2D)

Coordenadas locales

$$\mathbf{k}_L = \left(\begin{array}{ccc|ccc} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ \hline -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{array} \right)$$

Coordenadas globales

$$\mathbf{k} = \left(\begin{array}{ccc|ccc} A & D & H & -A & -D & H \\ D & B & F & -D & -B & F \\ H & F & C & -H & -F & C/2 \\ \hline -A & -D & -H & A & D & -H \\ -D & -B & -F & D & B & -F \\ H & F & C/2 & -H & -F & C \end{array} \right)$$

donde

$$\begin{aligned} A &= c^2 \frac{EA}{L} + s^2 \frac{12EI}{L^3}; & B &= s^2 \frac{EA}{L} + c^2 \frac{12EI}{L^3}; \\ C &= \frac{4EI}{L}; & D &= cs \left(\frac{EA}{L} - \frac{12EI}{L^3} \right); \\ F &= c \frac{6EI}{L^2}; & H &= -s \frac{6EI}{L^2} \end{aligned}$$

A.3. Emparrillado plano (E2D)

Coordenadas locales

$$\mathbf{k}_L = \left(\begin{array}{ccc|ccc} \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 \\ \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} & 0 \\ 0 & 0 & \frac{GJ}{L} & 0 & 0 & -\frac{GJ}{L} \\ \hline -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 \\ \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} & 0 \\ 0 & 0 & -\frac{GJ}{L} & 0 & 0 & \frac{GJ}{L} \end{array} \right)$$

Coordenadas globales

$$\mathbf{k} = \left(\begin{array}{ccc|ccc} A & D & -H & -A & D & -H \\ D & B & F & -D & M & P \\ -H & F & C & H & P & N \\ \hline -A & -D & H & A & -D & H \\ D & M & P & -D & B & F \\ -H & P & N & H & F & C \end{array} \right)$$

donde

$$\begin{aligned} A &= \frac{12EI}{L^3}; & B &= c^2 \frac{4EI}{L} + s^2 \frac{GJ}{L}; \\ C &= s^2 \frac{4EI}{L} + c^2 \frac{GJ}{L}; & D &= c \frac{6EI}{L^2}; \\ F &= cs \left(\frac{GJ}{L} - \frac{4EI}{L} \right); & H &= s \frac{6EI}{L^2}; \\ M &= c^2 \frac{2EI}{L} - s^2 \frac{GJ}{L}; & N &= s^2 \frac{2EI}{L} - c^2 \frac{GJ}{L}; \\ P &= -cs \left(\frac{2EI}{L} + \frac{GJ}{L} \right) \end{aligned}$$

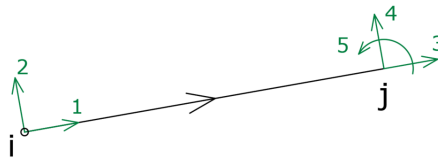
Anexo B

Matrices de rigidez elementales con gdl's condensados

En el presente Anexo se incluyen las matrices de rigidez elementales en coordenadas locales y globales para pórtico bidimensional en el caso de condensación de gdl's de giro o de desplazamiento perpendicular al elemento en un extremo.

B.1. Pórtico plano con libertad de giro en el extremo inicial

Se considera un elemento P2D con nudo inicial i y nudo final j . Se condensa el giro en el nudo inicial (θ_i).



Coordenadas locales

$$\mathbf{k}_L = \left(\begin{array}{cc|ccc} \frac{EA}{L} & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{3EI}{L^3} & 0 & -\frac{3EI}{L^3} & \frac{3EI}{L^2} \\ \hline -\frac{EA}{L} & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{3EI}{L^3} & 0 & \frac{3EI}{L^3} & -\frac{3EI}{L^2} \\ 0 & \frac{3EI}{L^2} & 0 & -\frac{3EI}{L^2} & \frac{3EI}{L} \end{array} \right)$$

Coordenadas globales

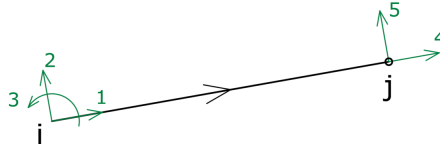
$$\mathbf{k} = \left(\begin{array}{cc|ccc} A & D & -A & -D & H \\ D & B & -D & -B & F \\ \hline -A & -D & A & D & -H \\ -D & -B & D & B & -F \\ H & F & -H & -F & C \end{array} \right)$$

donde

$$\begin{aligned} A &= c^2 \frac{EA}{L} + s^2 \frac{3EI}{L^3}; & B &= s^2 \frac{EA}{L} + c^2 \frac{3EI}{L^3}; \\ C &= \frac{3EI}{L}; & D &= cs \left(\frac{EA}{L} - \frac{3EI}{L^3} \right); \\ F &= c \frac{3EI}{L^2}; & H &= -s \frac{3EI}{L^2} \end{aligned}$$

B.2. Pórtico plano con libertad de giro en el extremo final

Se considera un elemento P2D con nudo inicial i y nudo final j . Se condensa el giro en el nudo final (θ_j).



Coordenadas locales

$$\mathbf{k}_L = \begin{pmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 \\ 0 & \frac{3EI}{L^3} & \frac{3EI}{L^2} & 0 & -\frac{3EI}{L^3} \\ 0 & \frac{3EI}{L^2} & \frac{3EI}{L} & 0 & -\frac{3EI}{L^2} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 \\ 0 & -\frac{3EI}{L^3} & -\frac{3EI}{L^2} & 0 & \frac{3EI}{L^3} \end{pmatrix}$$

Coordenadas globales

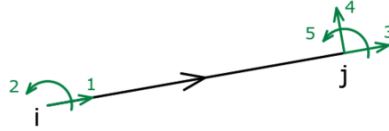
$$\mathbf{k} = \begin{pmatrix} A & D & H & -A & -D \\ D & B & F & -D & -B \\ H & F & C & -H & -F \\ -A & -D & -H & A & D \\ -D & -B & -F & D & B \end{pmatrix}$$

donde

$$\begin{aligned} A &= c^2 \frac{EA}{L} + s^2 \frac{3EI}{L^3}; & B &= s^2 \frac{EA}{L} + c^2 \frac{3EI}{L^3}; \\ C &= \frac{3EI}{L}; & D &= cs \left(\frac{EA}{L} - \frac{3EI}{L^3} \right); \\ F &= c \frac{3EI}{L^2}; & H &= -s \frac{3EI}{L^2} \end{aligned}$$

B.3. Pórtico plano con libertad de desplazamiento en el extremo inicial

Se considera un elemento P2D con nudo inicial i y nudo final j . Se condensa el desplazamiento perpendicular al eje del elemento en el nudo inicial ($u_{iy,L}$).



Coordenadas locales

$$\mathbf{k}_L = \left(\begin{array}{cc|cc} \frac{EA}{L} & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{EI}{L} & 0 & 0 & -\frac{EI}{L} \\ \hline -\frac{EA}{L} & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{EI}{L} & 0 & 0 & \frac{EI}{L} \end{array} \right)$$

Coordenadas globales

$$\mathbf{k}_L = \left(\begin{array}{ccc|ccc} \frac{EA}{L}c^2 & \frac{EA}{L}cs & 0 & -\frac{EA}{L}c^2 & -\frac{EA}{L}cs & 0 \\ \frac{EA}{L}cs & \frac{EA}{L}s^2 & 0 & -\frac{EA}{L}cs & -\frac{EA}{L}s^2 & 0 \\ 0 & 0 & \frac{EI}{L} & 0 & 0 & -\frac{EI}{L} \\ \hline -\frac{EA}{L}c^2 & -\frac{EA}{L}cs & 0 & \frac{EA}{L}c^2 & \frac{EA}{L}cs & 0 \\ -\frac{EA}{L}cs & -\frac{EA}{L}s^2 & 0 & \frac{EA}{L}cs & \frac{EA}{L}s^2 & 0 \\ 0 & 0 & -\frac{EI}{L} & 0 & 0 & \frac{EI}{L} \end{array} \right)$$

B.4. Pórtico plano con libertad de desplazamiento en el extremo final

Se considera un elemento P2D con nudo inicial i y nudo final j . Se condensa el desplazamiento perpendicular al eje del elemento en el nudo final ($u_{jy,L}$).



Coordenadas locales

$$\mathbf{k}_L = \left(\begin{array}{ccc|cc} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{EI}{L} & 0 & -\frac{EI}{L} \\ \hline -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 \\ 0 & 0 & -\frac{EI}{L} & 0 & \frac{EI}{L} \end{array} \right)$$

Coordenadas globales

$$\mathbf{k}_L = \left(\begin{array}{ccc|ccc} \frac{EA}{L}c^2 & \frac{EA}{L}cs & 0 & -\frac{EA}{L}c^2 & -\frac{EA}{L}cs & 0 \\ \frac{EA}{L}cs & \frac{EA}{L}s^2 & 0 & -\frac{EA}{L}cs & -\frac{EA}{L}s^2 & 0 \\ 0 & 0 & \frac{EI}{L} & 0 & 0 & -\frac{EI}{L} \\ \hline -\frac{EA}{L}c^2 & -\frac{EA}{L}cs & 0 & \frac{EA}{L}c^2 & \frac{EA}{L}cs & 0 \\ -\frac{EA}{L}cs & -\frac{EA}{L}s^2 & 0 & \frac{EA}{L}cs & \frac{EA}{L}s^2 & 0 \\ 0 & 0 & -\frac{EI}{L} & 0 & 0 & \frac{EI}{L} \end{array} \right)$$

Anexo C

Glosario

A continuación se muestra el significado de los principales símbolos y abreviaturas empleados en la presente obra.

- A : área de la sección de un elemento.
- A2D: abreviatura de «estructura articulada bidimensional».
- A3D: abreviatura de «estructura articulada tridimensional».
- \mathbf{A} : matriz de flexibilidad de una estructura.
- c : abreviatura de « $\cos \alpha$ ».
- E : módulo de elasticidad o módulo de Young.
- E2D: abreviatura de «emparrillado plano».
- \mathbf{F} : vector de fuerzas nodales de una estructura.
- $\tilde{\mathbf{F}}$: vector de fuerzas de empotramiento de una estructura.
- gdl: abreviatura de «grados de libertad».
- G : subíndice que indica que un determinado vector o matriz está en coordenadas globales. Generalmente omitido.
- I : momento de inercia de la sección de un elemento.
- \mathbf{I} : matriz identidad.
- \mathbf{k}^e : matriz de rigidez de un elemento e .
- $\mathbf{k}^{\text{muelle}}$: matriz de rigidez de un apoyo elástico o conjunto de muelles.
- \mathbf{K} : matriz de rigidez de una estructura.
- L : longitud de un elemento.
- L : subíndice que indica que un determinado vector o matriz está en coordenadas locales.
- \mathbf{m}_{AB} : matriz de cambio de coordenadas de un nudo de un sistema de coordenadas A a otro B .

- M_{AB} : matriz de cambio de coordenadas de un elemento de un sistema de coordenadas A a otro B .
- MDR: abreviatura de «Método Directo de la Rigidez».
- $_N$: subíndice que indica que un determinado vector o matriz está en coordenadas nodales.
- P^e : vector de fuerzas nodales de un elemento e .
- \tilde{P}^e : vector de fuerzas de empotramiento de un elemento e .
- P2D: abreviatura de «pórtico bidimensional».
- P3D: abreviatura de «pórtico tridimensional».
- s : abreviatura de «sen α ».
- SR: abreviatura de «sólido rígido».
- T : matriz de ligadura entre gdl.
- u : vector de desplazamientos de la estructura.
- δ : vector de desplazamientos de un elemento.
- u_i : vector de desplazamientos en el nudo i de la estructura.
- α : ángulo de giro de un sistema de coordenadas con respecto al sistema de coordenadas global (medido en sentido antihorario desde el sistema global).
